THE ADDRESS BUS

PORT TO PORTAL -- Editorial

I feel a bit like the proverbial phoenix, rising from the ashes of its prior life. Yes, I'm back! Although pulling this double-issue together took a month longer that I thought when I last corresponded with you! My apologies for the delay. But I have a few things to comment on before releasing you to peruse the rest of the issue.

SPARE SYSTEM. As I remarked on the postcard or letter you last received from me, I've brought up another '90. This is presently a two-drive, soft-sector setup and I'm now attempting to bring up a Z-67 hard drive I acquired a year ago as a add-on to the spare machine. Regrettably, one of the things that died when my primary system crashed last September was C.D.R.'s RAMdrive. I called C.D.R. to see if it could service the thing, but it no longer will! Since preparation of this newsletter virtually demands high storage capacity and fast access--my main use of late for the RAMdrive--the Z-67 is intended as a replacement for it. Further, the Z-67 would give me access to an 8-inch drive to support those of you who run that medium. So far, though, the '67 is being difficult! But at least the pair of internal, soft-sector drives are functioning and I've used them already while my main system has been occupied with something else.

ZUG LIBRARY. The most exciting news in the past year is that ZUG has released its 8-bit software library to **SEBHC Journal** and **STAUNCH**. Lenny Geisler picked up the disks directly from Pat Swayne, ZUG's former software engineer, and is presently organizing and duplicating them. I just received the first block of **60** disks from him and will begin listing this material in the next issue. Cost will be somewhat higher than my usual, but I'll also be abiding by the agreement Lenny made with Pat, when he acquired the library. A portion of the price will be passed back to Pat. That money (such as it will be) might well be plowed into the "new HUG" that the three principals of the old ZUG are trying to form. See VENDOR.UPDATE for details.

CALL FOR ARTICLES. My backlog of potential articles is beginning to thin. The regulars (Hank Lotz and Pete Shkabara) will continue to be here, I have quite a bit of material to cover myself--see THE ADDRESS BUS for hints of that, as well as my continuing series on software testing--and Lee Hart will join me next year in a discussion of T/MAKER. Still, my present supply of **new** articles will expire early in the new year. Although I could fall back to reprinting old stuff from **REMark**, I think a mix of old **and** new is important. So if you've been working on a computer project of late and would like to make a little money (yes, $$$$!) by writing about it, contact me. I'll send you my author's guide. Please **don't** send me a disk or manuscript without prior discussions with me, though.

ADDRESS LABELS. By the way, you may have noticed one change in my production this year. The address label on the envelope has been done on a dot-matrix printer. My old daisyweel printer crashed last December (as I mentioned in the last two issues) while running off labels. One of the labels curled over, jammed the roller platen, and that took out the circuit board that drives the platen! I had planned to discuss that situation and service for the Diablo 630 in this issue, but for want of space will have to postpone it. However, I've moved **all** label production to my C.Itoh 8510B 9-pin dot-matrix. Platen jams there will be less harmful to the printer!

RENEWALS. Finally, the subscription year is rapidly drawing to a close. Check the issue number on the label on this issue's mailing envelope. This "year" will close with a **double**-issue, #32/33. If the "last issue" number on the mailing label is greater than 33, you still have a year or two to go on your subscription. If that number **equals** 33, the next issue will be your last. But renewal rates are the same as this year, $15 domestic and Canada, $19 overseas. (Sorry, but there's a **typo** in the price listed at the end of the last issue! I've just corrected the file.)

Kirk L Thompson

THE EIGHT-BIT R/W -- Letters

Raves for Lee's Portable '89. [From Emil Riedel, Houston, TX] "Past information has indicated that a portable H-89 may be available some day. That sounds interesting to me. Hopefully, work is progressing on this idea."

[From Tom Slavik, Waco, TX] "I called Lee Hart last night to find out about the possible progress on the development of the notebook computer for the H-89. I have read letters of interest in **Staunch** and the **SEBHC Journal** for its development. Some of you may not realize the amount of time and money this project would entail. Lee informed me that time alone for development could be a full year for one person. I was willing to send a check for the cost of one notebook (which I figured to be around $200) to see

SOFTWARE LISTING
>---<

General Software Catalog

A catalog of **Staunch** software is available. Initially prepared by Ralph Money, the disk files include listings for **both** HDOS and CP/M. If you wish, the files can be "squeezed" to conserve disk space and an unsqueezer provided to recover them. This catalog now requires **four (4)** disks in standard hard-sector, one or two disks in most other formats if unsqueezed. See the "Placing an Order" section, below, for information on formats.

Source Code in this Issue

If this issue includes any source code, be it BASIC, Pascal, C, assembler, or whatever, you may obtain it **at no charge!** Merely send a **formatted** disk with a **postage-prepaid** mailer and I'll tranfer it for you. Please clearly indicated the format you are supplying. See below for supported formats.

FOR HDOS

Getting Started With HDOS and Assembly Language Programming: A Primer for the Beginner
By William N. Campbell, M.D.

This material is an introduction to HDOS assembler programming for the absolute novice. It includes use of the assembler, number systems, pseudo-opcodes, XTEXTs and .ACM files, and an introduction to HDOS SCALLs and the 8080 instruction set. The original hardcopy document that Doc Campbell prepared included references to Heath's old **Assembly Language Programming** course (EC-1108). With the kind permission of Heath Co. and its Product Line Manager, Education (Douglas M. Bonham), those sections are **excerpted** as an appendix in this document. Also included are quick reference tables to the HDOS SCALLs, 8080 opcodes, and numeric conversions; and a pair of elementary assembly language articles by Pat Swayne from early **REMarks**. Note, though, that this is **only** an introduction to assembler programming, with the emphasis on running your programs under HDOS. Other materials you will need are either chapter 7 of the HDOS 2.0 system manual ("HDOS System Programmer's Guide") or chapter 13 of the HDOS 3.02 manual ("HDOS Programmer's Reference Manual") **and** a formal text on 8080 assembler programming. The latter could include Heath's EC-1108 course (see this issue's CONTACTS column), Fernandez and Ashley's book reviewed by Hank Lotz later in this issue, or any of a number of similar books published during the early 1980's. This all-text-file package requires 1138 sectors. If desired, this package could be supplied on CP/M media, requiring 300K. My thanks to Heath Co. for permission to excerpt EC-1108 and to Dan Jerome for keying the material.

SMUGH Disk Libraries
(My thanks to William Cordes and Daniel Jerome, SMUGH librarians, for permission to distribute this material.)

Note: each of the individual disk listings below is **equivalent** to a one hard-sector (90K) disk. Disks will be listed five at a time per issue for each system until the library is complete.

FOR CPM ONLY

SMUGH CP/M DISK #1--Released: Jan 83

File	Ext	Size		Description
CAT	.COM	1K	\	These files are used to
CAT2	.COM	1K	\	create and read disk
CATALOG	.DOC	5K	\	catalogs, and are part
FMAP	.COM	4K	/	of a CP/M disk catalog-
FMAP	.DOC	2K	/	ing system. CATALOG.DOC
UCAT	.COM	2K	/	explains how to use the system.
FINDBAD	.COM	2K		Finds bad sectors on a disk.
FINDBAD	.DOC	5K		If it finds any, it creates a file "UNUSED.BAD" and prevents attempts to write to the bad sectors.
H89STAT	.COM	3K		Displays various statistics about the current state of your CP/M system.
SIG/MCAT	.001	60K		An early list of SIG/M disks.
UMPIRE	.COM	2K		A set of diagnostic routines that wrings out your system. Explains what is happening as it goes along.

SMUGH CP/M DISK #2--Released: Feb 83

File	Ext	Size	Description
CRCK	.COM	2K	Computes a CRC file in HEX for any file. May be used to verify modem transfer.
MODEM7	.AQM	32K	The standard CP/M modem program.
MODEM7	.COM	9K	Uses ACK/NAK protocol and transfers one sector at a time with checksumming. Binary file displays list of commands.
SQ15	.COM	14K	A file compression program.
USQ	.COM	10K	The reverse of SQ15.COM. It un-squeezes a squeezed file so it can be used.
XFER	.COM	4K	Computes the file transfer time at the desired baud rate.

SMUGH CP/M DISK #3--Released: Mar 83

File	Ext	Size		Description
HELP	.COM	2k		The main "Help" system program. Enter Help<RTN> or Help filename.HLP<RTN>.
ASM	.HLP	4K	\	Tells you all you ever wanted
CPM2	.HLP	37K	\	to know about these items,
HELP	.HLP	7K	/	and then some. HELP.HLP ex-
MBASIC	.HLP	21K	/	plains the system and how to initiate .HLP files.
SPELL	.COM	3K		A simple spelling check program.
SPELL-ED	.COM	9K		You create your own "lexicon.-
SPELL-ED	.DOC	4K		lex" file in one-word lines in upper case. You may expand your .lex file as you go along.

SMUGH CP/M DISK #4--Released: Apr 83

File	Ext	Size	Description
AMORTIZE	.BAS	7K	An amortization program.
DEPREC	.BAS	3K	A depreciation program.
ASSETS	.CAL	4K	A net-worth SuperCalc template.

CIA	.BAS	15K	\ A MBASIC adventure game
CIA	.DAT	3K	\ contributed by
CIA	.HLP	2K	/ Duane Klinge.
CIA	.SAV	1K	/
CRUNCHER.BAS		5K	Computes statistical data for up to 1000 data values and prints out a histogram chart (a 'Number Cruncher'). Contributed by Jerry Bloch.
LEDGE	.DOC	9K	A set of 9 programs which com-
LEDGEACT.BAS		3K	prise a double-entry bookkeeping
LEDGEBAL.BAS		3K	system. Enter MBASIC LEDGEMEN to
LEDGECRE.BAS		4K	start at menu. Contributed by
LEDGEENT.BAS		6K	Jerry Bloch - an Osborne user !
LEDGEMEN.BAS		2K	
LEDGENEW.BAS		3K	
LEDGEPRN.BAS		2K	
LEDGEPRO.BAS		3K	
LEDGETRI.BAS		3K	
PPONG	.COM	7K	A binary "pong" graphics game
PPONG	.DOC	1K	written by Mike Warren.

SMUGH CP/M DISK #5--Released: May 83

APPEND	.COM	1K	Appends one file to another
APPEND	.DOC	1K	file, but read the .DOC.
DU	.COM	6K	A direct access disk utility.
DU	.DOC	6K	BE VERY CAREFUL, you could ruin your disk.
FILE-XT2.COM		1K	A different type of directory
FILE-XT2.DOC		1K	display.
MEMCHEK	.COM	2K	A memory checking program. Con-
MEMCHEK	.DOC	1K	tributed by Bob Moffitt, a CRO-MENCO user !
RESOURCE.COM		8K	Ward Christensen's disassembler.
RESOURCE.DOC		14K	
SWEEP	.COM	26K	A file management program with
SWEEP	.DOC	1K	many options. Contributed by Jerry Bloch, an OSBORNE user !
WARMBOOT.ASM		3K	A program to replace CTRL-C.
WARMBOOT.DOC		1K	NOTE: must be assembled. Read .DOC first.
XDIR	.COM	10K	An extended directory program
XDIR	.DOC	6K	with many options. Contributed by Bill Flynn.

FOR HDOS ONLY

SMUGH HDOS DISK #1

NOTE: This disk was deleted by order of the SMUGH Board of Governors, effective October 1984. It contains commercial material, and this violates club by-laws, and the copyright laws of the United States.

SMUGH HDOS DISK #2

BANNER	.BAS	63	Prints a banner type message in
BANNER	.DOC	6	many different size characters.
CLOCK8	.BAS	9	BHBASIC clock program for the H8.
DOGFIGHT.BAS		17	BHBASIC game. Airplane dogfight game.
FLAGS	.ABS	5	A new flags program.
MAZE	.BAS	17	BHBASIC game. Travel through a maze.

N	.ABS	3	H8 disk-renaming program.
SMUG	.PIC	21	Banner printout of SMUGH.
SPCSHIP	.BAS	14	BHBASIC game. Shoot spaceship
SPCSHIP	.DOC	6	out of orbit.
SYSCMD	.SYS	9	A modified system file.

SMUGH HDOS DISK #3

CHUG	.TXT	22	Info from Capitol HUG.
CRAPS	.BAS	26	BHBASIC game.
DISKMENU.MBS		11	Displays disk menu while in
DSKMENU2.MBS		11	MBASIC (two versions).
HEATHER	.PIC	27	Poster..
MOUNTALL.ASM		11	Mounts all available disks.
NOSPAC	.ACM	18	Eliminates need to type spaces
NOSPAC	.ASM	14	on boot, HDOS VER 1.6.
OTHELLO	.BAS	24	BHBASIC game.
PHLAGS	.ABS	4	\
PHLAGS	.ASM	32	> Flag changing utility.
PHLAGS	.DOC	6	/
ROBOT	.BAS	10	BHBASIC game.
ROBOT	.DOC	3	
SALESMAN.ABS		129	Game from Software Toolworks (with permission).

SMUGH HDOS DISK #4

MORSEC	.BAS	19	H19 morse code simulator in
CIPHER	.DAT	4	BHBASIC.
CONTROL	.BAS	49	\ These 4 BHBASIC programs are
DISPLAY		13	\ used together for a 747 sim-
FLIGHT	.BAS	3	/ ulator. Runs very slowly in
INSTRUCT.BAS		42	/ MBASIC.
DSMCHAMP.ABS		19	As you would imagine, a disas-sembler.
HEAT81		59	1981 calendar with heather (graphics).
PDUMP	.ABS	16	ABS dump to LP: by filename.
PIRATE	.BAS	33	Adventure game called "PIRATE".
PIRBLD	.BAS	51	Run this BHBASIC file first to
PIRATE	.DTB	1	build up a data base, then run "PIRATE.BAS".
PONG	.ABS	3	For H-19. Uses keypad to con-trol ball.
SMUG	.PIC	3	Try to guess what this is.

SMUGH HDOS DISK #5

ASCII	.ACM	3	ASCII Character Equivalents.
BUNNY	.PIC	8	Posters are getting to be a tradition.
CURSE	.BAS	6	A funny little BHBASIC program.
DISKSLOW.ABS		1	A program that lets you play
DISKSLOW.ASM		2	with the motor shut off time on your disk drives.
DSKCASX	.ABS	6	A program that lets you write
DSKCASX	.ASM	50	out backups onto tape. (Not
DSKCASX	.DOC	12	checked out on the H89).
EPROM	.ACM	9	REMark Issue #13 has an article on building an EPROM board for an H8. This is a routine that someone else wrote for a dif-ferent EPROM programmer, so if you want an EPROM board, here is some help.
FUZZY	.BAS	14	If you can't make up your mind, try this BHBASIC program.
GO	.ABS	1	A simple PROLOGUE.SYS program

GO	.ASM	1	that can be modified very easily. Loads and runs any program that you want.
GO	.DOC	4	
GOBLIN	.BAS	48	Let's go on a treasure hunt, but watch out for the goblin. (He lurks in both BHBASIC and MBASIC!)
GOBLIN	.DOC	4	
HDOSX	.ACM	22	Defines HDOS Parameters.
INTRO	.ABS	2	Remember the first program in the H8 manual? 'YOUR H8 IS UP AND RUNNING (BEEP)' This program does not run on the H89, as it uses the front panel LEDs of the H8.
MELIZA	.MBS	20	Got a problem, ask MELIZA! (for MBASIC)
NAMEDISK	.ASM	12	An early disk renaming program-- Not a good one, but it works, and the source is here to play with.
NAMEDISK	.DOC	5	
OSCAR	.BAS	15	A BHBASIC program to help you find your friendly "local" ham satellite when you want to.
QUAD	.BAS	10	BHBASIC program to design your own quad antenna.
SUBMIT	.PAC	1	Patches to allow the HUG program SUBMIT.ABS to run under HDOS ver 1.6.

Note: A catalog to SMUGH's **entire** collection is separately available from the general software catalog. It requires 635 sectors of storage under HDOS or 160K for CP/M. If you're running hard-sector and want to conserve storage, ask for **compressed** files; I'll include an expansion utility and instructions with the files. Compression will reduce storage requirements by approximately one-half.

SOFTWARE UPDATE

TXT2Q3 for CP/M
(Originally announced in issue #'s 3 and 7)

Version 2.2 of this utility to convert ASCII files to Hoyle & Hoyle's QUERY!3 database format is now available. Though upwardly compatible with the various releases of version 1, this new version **integrates** the destination .DTB format into the text source file. Two on-line help screens are accessible from the main menu and the program is compiled. MBASIC and assembler source code is included. This version was prepared by Hank Lotz and is based on interpreter code written by Kirk Thompson. If you have any of the earlier CP/M versions of this program and want this update, **return** your **original** distribution disk for TXT2Q3 with a postage-prepaid return mailer and the files will be transferred at **no charge**. This material requires 76K of disk storage. A **compiled** HDOS version of this update is in the works!

Placing an Order

Your cost for this software depends on what you supply:

Formatted disk(s) and self-addressed, stamped return mailer .... $2.00 per disk

Formatted disk(s) without mailer .... $4.00 per disk
No disk(s) or mailer .... $6.00 per disk

Disk formats available are standard (SS/SD) or double-sided (DS/SD), 48-tpi hard-sector and single- or double-sided, 48- or 96-tpi soft-sector for both HDOS and CP/M. (**Staunch** now supports 96-tpi soft-sector whether you supply a **formatted** disk or not; see the lead article in the last issue.) If you do **not** have an H/Z system, **Staunch** also supports **many** CP/M-80 formats, such as Eagle II, Osborne, Kaypro, and Xerox, as well as PC-XT; 96-tpi support is provided through the courtesy of David McGlone of **The Z-Letter**. Let me know what your requirements are. Please **clearly** indicate the format you are supplying or require. If you desire DS hard- or any soft-sector format, I will pack multiple items onto one disk. I will **not** subdivide a disk. Send mail-orders to:

Kirk L Thompson / **The Staunch 8/89'er** / P.O. Box 548 / West Branch, IA 52358
>--<

THE EIGHT-BIT R/W (Continued from p. 1)

if it would help get things going. Lee came up with a better idea. Let's form our own company. We invest lets say a $50, $100 or whatever. This would give us shares in the project which could apply towards purchase. Persons who donate time toward the development of the system would receive shares for their efforts. Dividing the work load among 10 to 15 persons would make it much less of a time burden for any one individual. Clay Montgomery suggested the same thing and also suggested that the elimination of a disk drive would save a lot of design time. You could simply load all of your files onto your notebook from your '89 (and vice versa) and have this portable information readily available in your car, boat, golf cart, maybe even your office. Eliminating a drive would reduce power consumption considerably opening the possibility of using solar power for recharging.

"Lee also brought up the possibility of the development of a chameleon-type notebook that could emulate other CP/M computers such as Apple, Commodore, Kaypro, etc. leaving the software applications to be produced by those individual CP/M user groups.

"Hey wouldn't it be great for the H-89 group to come up with a development like this? Our group working together to create something of our own that could rival a new product at IBM. Whoa!! How many of you would like to show off your mini '89 to your PC friends and say: 'Yea, it's a new development. It's called a pocketbook, and it never needs recharging.'

"I have been a member of this group since November of 1989. I have read the letters of many of you that were upset that your computers had declined in value considerably. Actually, I envy the fact that I did not have the experience of building my first computer. I am very thankful that I bought this old dinosaur. The experience and knowledge that I have gained is immeasurable. Thanks to the infinite patience of folks like Kirk Thompson, Lee Hart and others, I now have an understanding of computers that I never would have achieved if I had entered the PC world. I can now share this knowledge with my

11 year old stepson. Want to spend quality time with a kid? Give him or her an H-89 and show them how to increase the T.L.B. to 3 MHz., increase baud rates, create SUBMIT command files, do some MBASIC programming, etc. How many parents have bought computers for their kids, given them a few game disks and felt that they have taught them something about computers. Staring blankly at a screen not having a clue as to what is going on inside the machine itself is not learning much about computers in my opinion.

"There is something neat about yanking the cover off my '89 and making a modification of some kind. It reminds me of my hotrod that I had in high school. The H-89 is a true classic. It was a machine ahead of its time at its introduction and still provides good service today. Please treat her with the respect that is her due and continue to support the efforts to keep her young. I like to think of her as a bright red '57 T-Bird. How many of you would like to trade your computerized, fuel injected, aerodynamic tin can for one? Boy I sure would!"

A Unified H/Z Publication? [From Bill Jacob III, Olympia, WA] "The following paragraph is from a letter I sent to Paul Herman of **Z-100 LifeLine**. I also sent a copy of it to Henry Fale at **H-SCOOP**.

"'Over the last year or so **Z-100 LifeLine, H-SCOOP, Staunch 8/89'er** and **SEBHC Journal** publishers have all complained about the loss of customers with the potential result of terminating the publication(s). I personally am on a tight budget and find the $84.00 annual renewal, for all the publications, hard to justify. I feel that Heath and HUG have finally done all of you a favor [see later in this issue for details]. If you can find a way to cooperate with each other, the demise of **REMark** magazine has opened the door to a new combined publication with sections from each of you. Add a section for a PC and you have the potential for a magazine that would outclass **REMark**, a volume to outdo your individual listings and at a cost saving to the subscribers. With the acceptance of ads and one shipping fee, the cost should drop well below the $84.00 individual subscription cost. I hope this idea is not too radical and that you all are already talking to each other about such a venture. Each of you fill a very different niche needed by many of us. I imagine a publication similar to the old **BUSS**, but far superior in that each of you is specialized.'

"Paul's immediate response was that people do not want information on all of the other computers, preferring to specialize on their own version. He stated that subscriptions have always declined in those situations. My feeling is that the drop in subscriptions was more due to the information on the old computers dropping to less than a page and included more as an afterthought than as serious information. I think the idea of a combined publication merits serious consideration." [I can sympathize with you, Bill, because I've been through a number of magazines over the years myself, culling out those that either didn't serve my needs or that I couldn't justify economically. Further, the only way I can foresee that such a publication as you propose could survive would be by **guaranteeing** each system ('89, '100, and PC) a certain number of

pages per issue. Certainly, one of the reasons for **Sextant** magazine's departure was the increasing paucity of 8-bit information back when the '8 and '89 were still plentiful. The introduction of the two specialist 8-bit newsletters, **SEBHC Journal** and **THE STAUNCH 8/89'er** BEFORE **Sextant** folded, illustrated that. However, given the proliferation and excellence of PC-oriented magazines, I have serious doubts whether the PC side of such a publication could have much to offer that doesn't **already** appear elsewhere. That would leave a combined magazine solely for the (out-of-production) '89 and '100. Would this have much appeal? Frankly, I concur with Paul Herman on this question, but what do the rest of you think? Could you put up with information about other systems if a section devoted to 8-bit topics was large enough? -Ed.]

Homebrew Electronics And Technological Hobbyist Computer Organization. [From Lee Hart, 323 W. 19th St., Holland, MI 49423; this letter arrived two months **after** ZUG bit the dust last summer. -Ed.] "I'm concerned about the rapid decline of ZUG (or HUG as I prefer to remember them). A user's group is by definition supposed to be run by users. It's time H/Z users took over again before it's too late!

"In the beginning, Heath was remarkably philanthropic. They generously contributed the money and resources to make HUG an exceptionally fine computer user's group. Yet they did not force HUG to grand special favors. Early **REMark**s routinely printed critical reviews, equipment modifications, and other information unfavorable to Heath, but valuable to readers. HUG programs helped us modify, expand, or avoid buying expensive software.

"Zenith was more sales oriented; they pressured HUG to turn a profit and promote sales of Heath/-Zenith products. Paid advertising began, and critical reviews went away. Articles focused on buying new hardware/software, not on ways to get more out of what you already had. HUG software that might compete with commercial products faded away.

"Matters have come to a head now that Groupe Bull has taken over. There is no place for a user's group in their grand strategic plans. In the end **REMark** was filled entirely by Bull and advertisers; nothing but a sales catalog you had to subscribe to.

"A user's group is absolutely vital now that all of us own an orphan computer. Do you want to see your multi-thousand dollar investment turn into junk for lack of parts or support?

"HUGgies unite! It's time to get our user's group back before it's too late!

"When Osborne failed, the First Osborne Group (FOG, their equivalent of HUG) formed a non-profit corporation to maintain support of their orphaned Osborne computers. They bought up the remaining parts, software, and manuals to insure a source of supply. They sponsored group buys, and did short run production of custom parts like ROMs, etc. I'm a member of two other clubs that did the same thing successfully: the Corvair Society of America, Inc., and the Tripoli Amateur Rocketry Association, Inc.

"I propose me make HUG a non-profit corporation. Its charter is to help members maintain and enhance their H/Z computers. It can continue the HUG library, bulletin board, and **REMark** magazine. It can

maintain sources for critical parts, like upgrade ROMs and hard-sectored disks. A good response will even let us continue to support those dedicated individuals who have contributed so much to the H/Z world (Pat Swayne, etc.).

"I even thought of a name—how about the Homebrew Electronics And Technological Hobbyist Computer Organization (HEATH CO for short)?

"How about it, guys! [Are there a]ny lawyers out there who know what it takes to set up a non-profit corporation? Do the members of ZUG have a right to vote on what happens to it? It's our money, too!" [Personally, I wouldn't mind the reestablishment of something similar to HUG, as Lee suggests. However, whether or not it's a non-profit corporation, the prime question is whether there's enough interest among existing users to sustain it. Declining circulation for those publications (**SEBHC Journal**, **STAUNCH**, and **Z-100 LifeLine**) that still support H/Z's orphaned computers suggests that there may **not** be! And I think we can mostly forget about supporting even the older PCs; that market already has a glut of glossy magazines and even the "new HUG" (ECI, discussed later in this issue) will, I think, have a tough time surviving. However, I'm only one voice. What do the rest of you think of Lee's resuscitated HUG? Like the notebook computer Lee has proposed, this kind of project requires **participation** by those who would reap its benefits. -Ed.]

Jumpering a Panasonic JU475-2 Drive. [From Charley Bejin, Cheyenne, WY] "My condolences on your hardware failure. The possible (inevitable?) failure of my steadily aging machines is a constant concern of mine. A port of my H-8 serial I/O board went out several months ago. Fortunately, a troubleshooting routine included with the Heath documentation enabled me to trace the problem to a bad chip. But by expertise is limited, and so is my time. Resources here in Cheyenne are also extremely limited, so a lot of transactions, etc., are accomplished via correspondence and mail order. That includes simple things like acquiring basic items such as chips. Unless one wants to make a two hundred mile round trip to Denver.

"Several weeks ago I wrote looking for some information about a Panasonic JU475-2 drive. Thought I'd share what I was able to come up with.

"The drive is [a] 1.2M 5-1/4". Jumper settings, as specified by ZDS for use in a Z248 series machine are DS2, DS, DC, CS, SP, AT, MM, & DA. DS2 probably depends on which drive, A:, B:, etc., you want to designate.

"I had been considering using it on either my H-8 or H/Z-89/90's. Due to its pedigree, however, it appears to be ideal for use externally with my SuperSport laptop. So that's where it went. I would suspect that a jumper setting or two may differ if it were to be used in one of the others." [Charley, you should probably consider yourself lucky to have used the Panasonic on your SuperSport. I'm told that a PC-style 1.2Meg drive, even though it's 80-track (96-tpi), requires **special** jumpering in order to use it in the '8 or '89. Specifically, the drive motor is designed to shift spindle rotation rate between 300 and 360 rpm. For use on our machines, the motor has to be strapped to run **only** at 300 rpm. I gather that if it were to run at 360 rpm, the

synch pulses sent to the controller by the drive would be too fast for the controller to handle. -Ed.]

Christmas in March. [From Alex Bodnar, Oxford, PA] "...I want to publicly T-H-A-N-K Mr. Jack Wert of Levittown, PA! He have me a call one evening in January and asked if I wanted any H-89 stuff because he wife gave him an ultimatum to get rid of all his 8-bit stuff. (As I am single, are most wives like that?) Of course, I asked him when [we] could get together and after getting there and to my surprise he 'GAVE' me F-O-U-R H-89s saying that 2 are working and 2 are questionable. I think that once I get settled in the new house I can get the other two going also. While I was thanking the Lord about all this, he throws in two MX-80 printers to boot! Christmas might have been December 25th but this day was MY Christmas!!! THANKX AGAIN JACK!!!"

Internal Foil Shielding. [From Arne Ostlund, Kungsbacka, SWEDEN] "Thanks for your letter and info. Glad to see that your system has been fixed, and that **The Staunch** will soon be in the mail again.

"Too bad if 'The old faithful' breaks down, especially here in Sweden where it is almost impossible to get assistance. I've been lucky so far, but must admit that I feel a sensation of fear every time I have to swing open the lid of the H/Z-90. A voice seems to whisper: 'Don't mess around with this machine or it will stop forever.'

"Had my share of problems when I recently installed the controller card for a Winchester drive. The card was too large. Fetched a hacksaw and divided the card into two. Mounted the parts and a home-made cable was used to restore the connections electrically. After that the computer was dead.

"Changed the wiring to a flat ribbon cable with every other wire grounded. Some signs of life, and I understood that the problem was noise-generated. Couldn't bring myself to desolder the cable a second time. Instead I got some aluminum foil from the kitchen and wrapped it around the cable. Also wrapped up the flyback transformer and grounded the aluminum 'packages.' That fixed it—the hard disk is now performing well without any problem. Must be a lot of radiation inside and around that machine!

"I should also mention that Mr. Darrell C. Pelan of Micronics Technology was very helpful when I mentioned my problems (although he didn't suggest the primitive aluminum-foil cure)."

Danish Chatter. [From Antony Phillips, Lyngby, DENMARK] "...I suppose by many people's reckoning I should give up my H-89s—I have two DOS machines, the latest a 486 for desktop publishing with Ventura (like you I produce magazines). However, I still have an H-89 loaned out to some one who types in material for the magazines on a disk (100K using PIE), which I tranfer on another machine to my DOS system. And I have another H-89 in the basement [which is] not working because the keyboard no longer connects to the CPU—I hope your series mentioned on one of your postcards will show me how to handle this, and also give me both courage and time to investigate the machine (see how much I want for my $32). I also have a comm port failure on the

machine here, but as I can use the printer port as a serial outlet, I don't have any problem transferring data. (I also have a stalwart H-125 printer.) And I have two sets of disks drives (700K) which don't seem to function when the weather is particularly humid.

"But all that is just chat ... I am curious to know how much longer you reckon you will keep the **Stauncher** going?..." [I replied, in part: "You ... asked how much longer I'll be publishing **Staunch**. That really depends on how much interest there is in the newsletter. This renewal go-around [winter '91/-92], circulation dropped **below** the **ad hoc** number for termination I had settled on when I took over the newsletter from Hank Lotz in '88. However, advance renewals from the committed readership have obligated me for another two (2) years (through 1993). I also want to do [some] promotional mailing[s] (like those postcards you've received) to see if I can rebuild circulation back above that **ad hoc** cut-off point. I have customer databases from three vendors who continue to support the '89, so want to do **at least** one mailing based on those before I throw in the towel. Unlike some other publications I could name, I plan to announce to the readership about a year in advance when that eventually [takes place]. (It **will** ultimately occur; the question is when. And at the moment, I frankly don't know!)"

[In addition, I should note that I've already done a promo mailing based on one vendor's customer list and have rebuilt the subscriber base to just above that cut-off point. When I've caught up with other things, I'll prepare mailings based on the remaining two customer databases. -Ed.]

Hardware Heaven? [From Dale Chidester, Salem, OH] "Enclosed is a check for my renewal. I may need [**Staunch**] for the hardware troubleshooting series. A local doctor gave me his Zenith Data Systems computers (Z-90s) and one H-89A. All were networked using the Corvus System and the Corvus 20MB hard drive. The internal hard-sectored drives were augmented by 8-inch DS/DD drives using Magnolia's 77316 I/O boards. The Corvus was controlled by Magnolia's 77314.

"One computer I altered by removing the hard-sector drive and its controller and installed the Magnolia 77316 in its place and [a] used external 8-inch drive. An FBE PIP [interface] enabled me to use the Centronics parallel connection on [an] IBM Wheelwriter [printer]. The recipient, a church secretary, is quite happy with the setup.

"On another one, the same procedure was followed but a 40-track drive, soft-sectored, replaced the hard-sectored drive. It is possible to install the Magnolia 77316 in P506/P512. My daughter has been using this one but it is used with a Brother daisy-wheel printer. She is using SuperCalc that was part of the [included] software. No computer problems yet.

"I bought a used but good H-37 controller board with installation software, but have met with disaster so far in a third computer. I thought I'd try Zenith CP/M; now I wish I had not." [Bringing up Zenith's CP/M on soft-sector can be a little tricky, Dale. I **re**discovered that when I set up my own spare system. If you'd like some help, let me know. -Ed.]

Anecdotal Service. [From Ronald Pannatoni, Franklin, NC] "I am writing in response to the comment in the last issue of **Staunch** [#28] that you are short on material to print. Consider this note a combined anecdote and 'service tip.'

"One morning back in November my H-89 'died.' When I switched the power on, the computer emitted just one "peep," and the screen stayed dark.

"Eventually I gave Henry Fale at Quikdata a call for advice. He told me that the monitor in the H-89 generates the 'peeps' as it checks the TLB and CPU board: one 'peep' for each board that is working.

"Thus, one of the boards had gone bad, but I needed additional information to determine which one.

"In fact, the TLB was the culprit. The clue that pointed to it was the dark CRT. When I opened up the H-89, I saw that the filament of the CRT was dark.

"Now the filament of the CRT in the H-89 gets its power by a very roundabout method. First, a sinusoidal signal originating on the TLB is passed to the video board by two cables that are routed through the CPU board. This signal is amplified on the video board and applied to the primary winding of the flyback transformer. The voltage for the CRT filament is taken off a secondary winding on this transformer (the winding is the wire that wraps around the outside of the transformer).

"Apparently, the TLB had stopped putting out its signal, so the flyback transformer had stopped putting out any voltage to the CRT. Replacing the TLB brought my H-89 back to life. (I gave Henry the bad TLB, but I have no idea what was actually wrong with this board.)"

=====

This 'n' That
by Hank Lotz / 2024 Sampson St. / Pgh, PA 15221

Hi to all! It's really great to be back after an absence of 2 issues! Here are 2 unrelated topics:

Another MBASIC Routine: Here's one to aid you when you type tables made up of columns of numbers (or words). This came about when, using an editor, I was preparing a grade sheet for a teacher friend. The sheet required 9 columns across the page (names and a lot of numerical grades). At 12 chars per inch on the printer you easily have room for 96 chars, but on the screen there's only room for 80. Therefore, on the screen each line wraps around, making it miserable to see what you're doing when entering the data. On top of that, if you **later** want to **change the spacing** of the columns (the "layout"), you have to add and delete spaces on every line!

So what I did was enter the numbers with **one symbol** (|) between them **instead of multiple spaces.** Then afterwards I employed a little MBASIC program to interpret those symbols as "tab stops" (changeable ones at that!) Not only was it a much easier job **typing the data**, but it was easier to **change the layout** -- which, as you must know, does end up needing to be changed more often than not! (Relying on editor tab sets doesn't provide special help for future changes. Nor does it get rid of the wrapping of lines on the screen.)

Here's the little program (LAYOUT.BAS), followed

by an explanation:

```
10 DEFINT A-Z
11 INPUT"Name of file with your tab symbols -- ";F$
12 IF INSTR(F$,".TAB")=0 THEN PRINT"Filetype must
   be .TAB" : GOTO 11
13 OPEN "I",#1,F$
14 Z=INSTR(F$,".") : G$=LEFT$(F$,Z)+"PNT"
15 PRINT "Creating file ";G$ : OPEN "O",#2,G$
16 IF EOF(1) THEN PRINT"END INPUT FILE": PRINT: END
17 B$="" : LINE INPUT #1,A$ : L=LEN(A$)
18 FOR J=1 TO L
19 C$=MID$(A$,J,1):IF C$<>"|" THEN B$=B$+C$:GOTO 22
20 READ T : PRINT #2,B$;TAB(T);
21 B$=""
22 NEXT J
23 PRINT #2,B$ : RESTORE : GOTO 16
24 REM -- TAB COLUMNS FOLLOW:
25 DATA 1,19,28,37,46
```

Line 25 is where you put your tab columns. You may use as many as you need, and change them at any time. The numbers you see there now are for a simplistic **example** which we'll now look at. Type your data into the editor as follows. (This happens to be a "grade sheet," too -- a fictitious one.) Note the so-called "tab stop" chars (|):

```
|NAME|EXAM 1|EXAM 2|EXAM 3|EXAM 4

|Clyde|100|95|100|99
|Jane|90|100|92|100
|Robert|75||62|79
|Melissa|100|99|100|100
```

Call this file GRADES.TAB. Notice I suffix all **input** files with .TAB, and the program gives all **output** files the extension .PNT. If you don't like that, change lines 12 and 14. But we gain **order** by that convention because it makes the I/O file relationship very clear. Plus, it's a **safer** procedure than just picking filenames out of the air at the risk of typing some unrelated filename in and overwriting it.

Now, running the program on GRADES.**TAB** gives this **output file**, GRADES.**PNT**:

NAME	EXAM 1	EXAM 2	EXAM 3	EXAM 4
Clyde	100	95	100	99
Jane	90	100	92	100
Robert	75		62	79
Melissa	100	99	100	100

Now that you can **see** what the program did to the input file, you're better able to grasp what I've been talking about all this time.

But alas, suppose this output is a first draft! It needs work. Let's say we want to indent the NAMES column over to column 8. All we have to do is change the first value in line 25 (the "1") to whatever column we want. So we make it an "8". Line 25 is now:

```
25 DATA 8,19,28,37,46
```

Also, let's take this opportunity to right-justify those grade columns. For this we **do** have to edit the source file (because LAYOUT.BAS is given here as a general-purpose program). Our new input

becomes:

```
|NAME|EXAM 1|EXAM 2|EXAM 3|EXAM 4

|Clyde|100| 95|100| 99
|Jane| 90|100| 92|100
|Robert| 75|| 62| 79
|Melissa|100| 99|100|100
```

We can now rerun the program. The new output is:

NAME	EXAM 1	EXAM 2	EXAM 3	EXAM 4
Clyde	100	95	100	99
Jane	90	100	92	100
Robert	75		62	79
Melissa	100	99	100	100

Notice that the first column moved **independently** of the other columns. If you had edited the input file instead of line 25, **everything** would have shifted to the right and then you'd have to move the other columns back (line by line!)

Some miscellaneous remarks. To make the typing of data easier, you **can** line it up on the screen **somewhat**, for you may use spaces (or even editor tabs) **before** each | mark -- so long as you don't ever type a | **beyond** whatever column you desire it to represent in the output! Also, note the use of the double tab where poor Robert didn't show up for Exam 2!

A Great Book to Teach You Assembly Language: I must preface this mini-review with two big qualifications: First, this book is good for CP/M, not HDOS. Second (and not least), it's been out of print for two years! (But see the closing paragraph below.) The book is **Introduction to 8080/8085 Assembly Language Programming**, by Judi N. Fernandez and Ruth Ashley, 1981, John Wiley and Sons, Inc., paperback, 303 pp, $12.95.

Don't let the 8080/8085 scare you; the language covered is compatible with the Z80 (under CP/M) in an H-89.

I can't conceive of a clearer book if you want to learn assembly language. It's one of Wiley's "Self-teaching Guides," and tests you at short intervals (frames) on what is presented. Frankly, I was never too crazy about that type of format for a textbook, but it did not get in my way here. In fact, this is invaluable for seeing if you really understand as well as you think you do!

The style of the book is relaxed, informal, and nontechnical. There are 12 chapters, 4 appendices, and a thorough index. There's a chapter on binary and hex number systems and their arithmetic. Other chapters discuss the stack, the registers, flags, subroutines, addition, subtraction, multiplication, division, BCD conversion, handling negative numbers, and much, much more! But so that you won't be intimidated by that description, let me add that I got an excellent handle on assembly language and still haven't read the whole book. To say I have more to learn is an understatement, but the book laid the foundation, and the book will supply plenty more when I'm ready.

Of course the opcodes are covered (as are assembler directives) and explained in detail, and

short programming examples are given to illustrate usage. And here is a valuable tip: Keep your Heath Operations Manual open to its "Instruction Set" section (the pages listing the opcodes) when reading the book under review here. You'll find this book makes the Heath Manual more comprehensible, while the Manual nicely complements the book's opcode discussions. Used together they're dynamite.

Some years ago I saw a short review of this same book in a magazine (I forget which one). As I recall, it criticized the book as being written in such a way that the reader had to read it again to learn the material. Personally, when using the book, I was too busy **learning** to notice that! And even if true, why harp on it when assembly language is something that has to be read multiple times **anyway** in order for it to sink in! Besides, you can later go back and use the book as a reference (I do this -- it is copiously indexed!). I don't exactly call that "rereading the book."

In a discussion of whether I should even bother to **review** this book (in view of its out-of-print status), **Staunch** editor Kirk Thompson commented: "...I think a review...would [still] be useful.... We're approaching the point where a used bookstore and the out-of-print services may be the **only** way to locate some of the CP/M-oriented books we want." Kirk has a mature outlook there. To ignore a good thing because it's scarce is out of character for us staunch recognizers of **quality**. For, surely, along with our appreciation of our so-called "obsolete" computers comes a view of "peesees" as a far rougher fate than scarcity of printed matter. Rather than surrender quality, we should welcome the hope-that-is-our-**only**-hope for obtaining some material. [I presently have a book search firm looking for copy of this book for myself. I'll let you know when I hear anything. Other books on assembly language programming may be found at your local public and university libraries, used bookstores, or Lee Hart's mail-order service announced in the last issue. -Ed.]

=====

CONTACTS
(A Wanted/For Sale/Swap Column)

Tom Slavik (5205 Links Drive, Waco, TX 76708, 817/754-5164) "I have ... included a list of stuff I have for sale. You may list it if you have room or instruct your readers to send [an] S.A.S.E." [Below are highlights from Tom's list; for the whole page and a half listing, send Tom an envelope. -Ed.]

CP/M SOFTWARE

On Hard-sector:

CP/M 2.204 DISTRIBUTION DISKS	$ 3.00
EASY APPOINTMENT/MANUAL	$10.00
BIOS-80 DISKS /MANUAL	$ 5.00
MYCHESS/MANUAL	$ 5.00
MBASIC-80 /MANUAL	$ 8.00
SUPERCALC/MANUAL	$15.00
REMBRANDT/MANUAL	$15.00
HARD-SECTOR DISKS(generic/mixed)	.50 (10 min. order)
HARD-SECTOR DISKS (Dysan) NEW	1.00 (10 min. order)

On Soft-sector:

INSTRUMENT FLIGHT/MANUAL	$10.00
LONG & LOUD/MANUAL	$15.00
REMBRANDT/MANUAL	$15.00

I will convert soft to hard and vice versa at no additional charge, CP/M only.

HDOS SOFTWARE

On Hard-sector:

PIE FULL SCREEN EDITOR/MANUAL	$ 3.00
TEXT/MANUAL	$ 3.00
INTRO. BASIC PROGRAMMING/MANUAL	$ 3.00
UD.DVD/MANUAL	$ 5.00
TOUCH-TYPING TUTOR/MANUAL	$ 2.00
HUG DISC XIV	$ 3.00

HDOS Games

MUNCHKIN	$ 3.00
SPACE PIRATES	$ 3.00
GALACTIC WARRIOR	$ 3.00
ADVENTURE	$ 3.00

HARDWARE

80-TRACK D.S. 96-TPI DRIVE	$45.00
"HERBY" DYNAMIC RAM DRIVE	$30.00 expandable
to 1 full meg (software included)	
GREEN CRT	$25.00

SHIPPING AND HANDLING:

PROGRAM DISKS: 1-3	$1.50 total
3 or more	free
CRT or DISC DRIVES	$5.00 each

Tony Venticinque (536 Redwood Rd, Bolingbrook, IL 60440) "...By the way, one of my systems uses a H-29 terminal. The H-89 CPU board is housed in a H-89 case with the video board and terminal board gutted out. Do you know of a way I can reset this system without turning off the power switch?"

Arne Ostlund (Backsippegatan 14, S-434 46 Kungs-backa, SWEDEN) "A few months ago you kindly informed me of three Winchester drive suppliers. I purchased a hard disk controller and interface card from Micronics Technology. Experienced some installation problems because of noise inside the H/Z-90. Had to wrap up the flyback transformer as well as a flat ribbon cable in aluminum foil. Primitive shielding-- but it works.

"I need some software to get the Winchester organized. At the moment I'm using the ZCPR2 (limited public domain), and would like to find out whether it is worth paying for a full system ZCPR 3.4-Autoinstall and perhaps also ZSDOS...."

James L. Dunn, Jr. (11 Page One Dr., Longview, TX 75601, phone: 903/757-5343) Jimmy is looking for hardware and/or software that will "keep time, date and year when [his '89] is turned off or at least can set when it is turned on. I had some idea from back when that this could be used to show when files were last used, date letters and any other use I could find for it."

Biff Bueffel (19820 NW Metolius Dr., Portland, OR 97229) "Several newsletters ago, I wrote you that

Rick Swenton and I had converted the H-89 BIOS to Z80 mnemonics and optimized the code where possible to gain the largest possible TPA. At that time the BIOS could only be assembled with an SLR-type assembler and linked with SLRNK+ or DR's LINK. The BIOS source has since been modified slightly so it will now assemble with ZASM, a public domain assembler. PREL.COM and a modified MAKEBIOS.COM are used. In other words, any H-89 user can now use our modified source files to create a custom BIOS.SYS.

"A BIOS.HDR is edited to set the drive types and include or exclude desired options (e.g., why waste memory with ETX/ACK routines if you are not using them). If you have a C.D.R. SuperRAM89, the driver can be made internal to the BIOS.

"If you can run MAKEBIOS to make a new BIOS.SYS, you know enough to use these files. I can offer BIOS support for EMULATE and BIOS-80 but I cannot (legally or ethically) supply you with the necessary support files. The source code is fully commented.

"The files require 260K of disk space (170K if crunched). My preferred disk type is the H-37 96-tpi (5-1/4" or 3-1/2") but I can arrange to write to H-17 90K drives. Double-sided 48-tpi drives are possible with difficulty.

"If you want this improved BIOS send me a formatted disk(s) in a mailer with return postage."

W.K. Clarkson (1612 Via Arriba, Palos Verdes Estates, CA 90274) "I have been a sometime subscriber to **The STAUNCH 8/89'er.** I have a vintage H-89, much modified and improved, that I am now offering for sale. It's a little bit like taking your dog in to be put to sleep--I built that computer, I enhanced it, I used it to do things my PC friends marveled at, ... But the harsh reality is that we use MSDOS computers (and that other M-brand) at work, and I find I must have a compatible computer a home to get real work done. I don't use the H-89 much any more, have gotten very rusty with it, and so, it is time...

"To be specific: I have a circa 1980 64K H-89, meticulously built by me, subsequently modified for the sturdier power supply, and many other enhancements as specified below. It still works perfectly. Along with it there is the H-77 dual external drive system, H-14 printer (does anyone still want one of those?), an extra Siemens FDD100-5B disk drive, C.D.R. Systems SuperRAM89 (512K), McGaffey "Quick-P" parallel printer interface board, modified cooling system, Micronics Technology 4 MHz "Speed Mod" (with corrected HDOS and CP/M software by me), fabric glare screen, ...

"Software includes HDOS, CP/M, MicroSoft FORTRAN for HDOS, and numerous Software Toolworks programs (PIE, TEXT, SPELL, C/80 compiler, RATFOR precompiler, PACK & CRYPT, games and other miscellaneous software). Also included are programs developed by me, and sold commercially in the dark ages, such as S-BASIC and DISKSORT, as well as numerous utilities. There are also many hard-sectored disks.

"...This is simply too good a system to relegate to the junk pile!"

Jared Freeman (5919 Megan Dr., Hamilton, OH 45011, phone: 513/844-6332 after 6 pm or 513/867-4934 days) "Please list the following in the wanted/for sale/swap column of the newsletter:"

FOR SALE: H-89 with 64K, 20 Meg hard drive, 1 40-track and 1 80-track drive and clock. Software included is ZSDOS operating system, Magic Wand, SuperCalc, Max-Plus, WordStar 4.0, DataStar, Pascal, Query!3 and misc. other files. 100% working order. H-19A terminal also in working order.

"Make an offer."

G. Emil Riedel (15511 Seahorse Dr., Houston, TX 77062, home: 713/486-4559, work: 713/474-6782) "...I bought several H/Z-89s and dual hard-sector drives at a school auction. One of the machines also has an 8" drive (H-47). I also got the documentation for most of the equipment and for a lot of software. Some of them work and some do not. I would like to get rid of all of this equipment for a little $$ or free if necessary. Please include a note to this effect in the next issue."

H-89 (or Z-89) working	$50
H-89 (or Z-89) not working	$20
Dual 5-1/4 drives (H-17)	$50
Dual 8" drives (H-47)	$100

"Software and documentation available also. Buyer pays shipping...."

Gary S. Melander (460 Garrison Place, Virginia Beach, VA 23452, phone: 804/498-9850) "I would like to make some of my equipment available to your subscribers at a reasonable price. Would you be so kind as to advertise the following items?"

H-8 computer with Zenith Z-80 CPU (and 3 MHz mod), H-17 and H-37 controllers, 4-port serial board, and DG 64K dynamic RAM. $100.00 including UPS ground shipping. Will also sell parts individually: Z80 CPU, $45.00. H-17 controller, $15.00. H-37 controller, $45.00. Serial board, $25.00. Memory board, $35.00.

H-8 chassis with gold-pinned motherboard, fan, power supply, and front panel. $40.00 including UPS ground shipping.

H-8 8080 CPU board with Z80 chip and adapter installed. Will work just like a Z80 board at 2 MHz, including the capability to use an H-37 soft-sectored controller. $25.00.

WH8-64 dynamic memory board. $35.00.

2 Full-height double-sided, 80-trk/side (96-tpi) floppy drives. $30.00 each, both for $55.00.

Hard disk controller set for a Z-100/Z-110/Z-120 computer. $75.00.

Z-120 for parts. Does not have any disk drives. Good working display, power supply, and floppy disk controller board. $65.00.

1-908-741-2513

Charles Hansen (22 Gloria Court, Tinton Falls, NJ 07724) "I was a **STAUNCH 8/89'er** until 1990 when I moved to the clone world to maintain commonality with my work computer (I know, boo! hiss!)... the H-8 and H-89 were fantastic equipment.

"Anyway, I have a lot of H-89 equipment and software which I would like to get rid of, hopefully to some H-89'er close by who could pick it up--I don't think the hardware would survive shipping. A list is given below. I want to get rid of it as a package deal--best offer gets it all. All software

original disks and manuals are provided, as well as all hardware manuals for computers and printer. Bootable disks for all three H-89 hardware configurations included."

HARDWARE

H-89 64K, two internal 640K DSDD H-37 5-1/4 drives, SuperRAM89 with 512K, ANAPRO 4 MHz kit ... (not working).

H-89 64K, two internal 640K DSDD H-37 5-1/4 drives, Magnolia RAMdisk with 128K, ANAPRO 4 MHz kit... (also not working).

H-89 64K, with one internal 320K 5-1/4 drive, two outboard H-77 640K 5-1/4 drives and 1 H-17 5-1/4 hard sector drive... WORKING.

Diablo 630 daisywheel printer... with printer stand and tractor feed... (not working). Lots of film (not cloth) ribbons.

CP/M SOFTWARE

CP/M 2.2.03 and 2.2.04, WordStar 2.26, Electric Webster 1.8a, EP/STAR 1.0, SuperCalc 1.05, BASIC-80 5.21, BASCOM compiler 5.3, dBASE II 2.4, SuperRAM89 1.3, CPC, Magnolia RAMdisk, Turbo Pascal 2.0, Calligraphy II 1.21,... [various HUG disks,] Derby Utilities, CompuMagic.

Edward Snow (Chair, Computer Information Science, Orlando College North Campus, 5500 Diplomat Circle, Orlando, FL 32810) [**Note:** this correspondence was initiated by a letter from Nancy Black, "Fearless Computing" columnist for **Computer Monthly**. She enclosed the following with her note:] "Dear Ms. Black, I am writing in response to your column in the June issue of **Computer Monthly**. In your column you indicated that you have, on occasion, knowledge of persons who desire to place used CP/M machines in worthwhile institutions. I am writing to request that you help me locate a machine in good usable condition for the Orlando College.

"I have been trying for some time to locate a CP/M-based system to use in several of the classes the college teaches in my department. For instance, in our Introduction to Computers class we trace the development of personal computers from the Altair to the MacIntosh. We discuss the Osborne, for instance as well as CP/M. Also, in our Operating Systems class CP/M is discussed. I would like to have a machine to use as a hands-on illustration of CP/M, as well as an example of the early portable computers. If you know of someone who could provide one for the college please have them contact me...

"I believe that new users are immediately caught up in the 'more is better' philosophy, and are coerced into purchasing equipment far more powerful and expensive than they need. But if they don't get a 386 processor, 120 megabytes of disk storage, 2 megabytes of primary storage, a VGA color monitor, a math coprocessor, etc., they are made to feel that they will be unable to do productive work. In reality, how many bytes do you need to store a term paper? How fast does a machine have to be to calculate a home budget? Most of the students in the introductory courses are amazed to find out that you can do all of these easily with a machine that has 64K of RAM and a 1 megaherz chip (Apple II)...."

[I wrote to Prof. Snow as follows:] "I just re-

ceived a photocopy of your letter of 1 June from **Computer Monthly** columnist Nancy Black. Although I cannot assist you with your immediate need for a portable CP/M computer system, you might be interested to know that support for CP/M-based systems exists. However, most of that support is from hobbyists such as myself.

"As an example, I'm enclosing the most recent issue of the bimonthly newsletter I publish, **The STAUNCH 8/89'er**. I specialize in the Heathkit H-8 and H-89, as you will see as you browse through the issue. In particular, I would draw your attention to my new 'R/O Media' column beginning on p. 11 [for other resources]..."

[Prof. Snow replied:] "...Perhaps one of your readers would be able to help in my search. Also, I would be pleased to receive a homebuilt such as you Heath H-8. The entire purpose of the college getting a CP/M machine(s) was to properly illustrate the history of the Personal Computer. What could be more personal than building your own PC? I am not looking merely for an Osborne or a Kaypro, but for machines that will help our students better understand the impact of the personal computer on our society, and how the PC got where it is today. Besides, being somewhat of a 'dinosaur' myself (having started programming on an IBM 1401 in the mid-60s) I prefer the early PCs to the techno-marvels we use today."

The Staunch 8/89'er (P.O. Box 548, West Branch, IA 52358, phone: 319/643-7136) I have a lot of stuff I'd like to dispose of. Most of it was passed to me by Dan Jerome. He acquired much of it when the Minneapolis/St. Paul Heathkit Electronics Centers closed several years ago. Dan also decided to clear odd software and other items from his own collection that he no longer expects to use. My thanks to him for his generosity.

For the complete page and a half list write or call weekday evenings or on weekends. There's entirely too much to list here. Included are boards, cables, disk drives, miscellaneous hardware and chips, an '89A cabinet, **REMark** and **Sextant** back issues, manuals, some CP/M software, and **three** printers.

Free Diskette Sleeves: Further, if you need good-quality 5-1/4 disk sleeves, I've acquired a bunch from various sources. If you need some, I'll mail them to you at your request in lots of 25 for free!

=====

Pete on CP/M
By Peter Shkabara

As usual, there seems to be a long delay between when the start of a project and the finish. I started writing this installment at the end of November (at least it was 1991). And here it is now the middle of February of a new year as this is being completed.

I had previously asked for feedback on what to write about [see issue #26/27, p. 5] and suddenly two letters arrive. Kirk Thompson asks about the possibility of an MSDOS virus infecting a CP/M machine and also about adding a fourth drive to the H-89. Then L. Van Hemelryck writes to ask about the

particulars of how EMULATE handles the different drive parameters in the BIOS. The drive questions are related and I will spend some time discussing them but first let's look at the virus question.

Viruses. My first reaction to Kirk's virus question was that there is no possibility of cross infection between a Z80/8080 machine and a 80x86 system. [Infection of a CP/M machine by a PC virus **is** remote, certainly. See issue #28 (p. 11) for consequences if one should. -Ed.] On further thought, however, there is an area where some problems may arise. For those who may still be unfamiliar with viruses in computers, a short discussion may be in order.

As in other aspects of life, there are people out there who gain satisfaction from causing suffering to others. A pyromaniac sets fires, a rapist attacks women, and all too many write graffiti on walls, scratch car paint and smash windows just for "fun". In the computer world, we have programmers who find pleasure in causing the destruction of disk files of other computer users. This is done in a number of different ways.

A "Trojan" program may contain a hidden "hot key" that activates the erasure of files on a disk. Other programs simply do destructive work when they run. These programs are disguised as some enhanced directory lister or game in order to entice an unsuspecting user to run it. But the most dangerous and destructive creation is the virus.

In biology, a virus invades a living cell and eventually causes its destruction. A computer virus is very similar in behavior. This is a program which installs itself into a program and then replicates itself by sending copies into other programs whenever the "infected" program is run. Eventually, there is some point at which the virus will begin destructive action such as the erasure of files or the formatting of a hard disk.

Getting back to Kirk's original question, "can a PC program infect a CP/M computer?" (or vice versa). In general, the answer would be, no. A program written for a Z80 or 8080 microprocessor will not run on an 80x86. However, there are emulators which make it possible to run CP/M on a PC. If this was done and a CP/M program contained a virus, it could very well infect the PC system. Destructive action could be possible as long as the PC retained the capability to emulate the CP/M environment.

Fortunately for the Huggers, the virus problem is mainly a PC and Mac problem, although UNIX systems have also suffered. I am not aware of any such activity under CP/M. At least for now, y'all should be safe!

Note that not all viruses are meant to be destructive. There was a well meaning Christmas message sent a few years ago which caused quite a bit of damage to university computing networks. Someone in Europe sent out a self replicating message which would wish everyone on the network a Merry Christmas and Happy New Year. The message would then replicate itself to be rebroadcast to all other nodes connected to the receiving site. This was all fine and dandy, but there was no provision for terminating the sequence. The end result was that an entire communications network, including many universities and research centers were knocked out of commission due to tied up communications lines. All computer systems had to be shut down and restarted to get rid of the "message gone wild". All this at a large dollar cost as well.

Migration. Now on to other matters. Some of you may start to migrate to newer computer designs. It might be a PC or a Macintosh. Here are some pointers which reflect my views on the subject. Before I go further, please realize that the H-89 was a fine computer. It had done a good job for me as have many other H-89s for their owners. However, just because it is still doing a good job is not sufficient justification to avoid upgrading. Take a good look at the newer technologies. Isn't there something out there which could do the job better?

In my view, the Apple Macintosh with its Motorola 680x0 processors is a far better design than anything that IBM and clones have come up with. The Mac had a great display monitor back in 1984. It took IBM till 1989 to begin to approach that quality. The only problem with the Mac that I see is in the form of $$$$$$. Many of those green papers are needed to pay for a Mac. At this time, the Mac is about twice the price of an equivalent-capability IBM clone. Of course, the word "equivalent" may be raised into question. The Mac **is** a better machine, but we need to measure the amount of work per dollar invested. At least that is the way I decide on an investment. Also, because the PC **is** the more popular machine, there is much more inexpensive software available for it than there is for the Mac.

Although there are also the Commodore Amiga (a fine machine) and the Atari ST series, these computers are a very small minority in the software arena. Both of these machines offer low cost entry into great graphics, but again, the breadth of software is small.

Migration from an H-89 to a PC can be made relatively painless by the use of a Z80 emulator or even a co-processor board. When I first drifted over to the PC, I could not give up my Z80 software. To solve the dilemma, I purchased a Z80 board from Micro Solutions. This board comes with CP/M emulation as well as the famous UNIFORM program. With UNIDOS, you can run CP/M programs interchangeably with any PC programs. The whole thing is completely transparent to the user. The only "penalty" is that you will loose a relatively small amount of TPA (transient program area).

UNIFORM is very nice. It allows you to designate one of your drives to emulate a particular CP/M format. This drive can then be accessed as if you were on a CP/M computer. For example, if you choose drive B: for emulation, then access to B: would DOS as usual, but you could also access it as drive E: which would then be a Heath CP/M format! You can log onto the E: drive and issue COPY, DEL, TYPE or other commands which may be appropriate. A great product design. Its only flaw is that support for the variety of Heath formats is not all that good. In addition, Micro Solutions' attitude for support has been very poor. I sent them a complete package of sample disks about four years ago. My package included complete with details of the disk formats as well as an offer to help them. The response was that if I paid them several hundred dollars, they would look at making the changes.

If any of you have the Micro Solutions UNIFORM package, I did develop some patches for the data file. These patches provide support for the Heath extended density formats. Let me know if anyone is in need of them.

Another product to consider is shareware. It is called 22NICE and 22DISK. Together, these packages provide Z80 emulation as well as disk format conversion. You may copy CP/M files to and from PC format. The whole process does not work anywhere as smoothly as the Micro Solutions package, but it is another alternative. And it **is** cheaper.

Now to look at the hardware front. At this time, it is possible to get an IBM compatible computer using a 25MHz 386SX processor with a VGA color display, 80MB hard disk and some bundled software for around $1200. Not a bad deal. There are some advertised "bargains" selling off 286 microprocessor based computers. In general, these bargains are not a good investment. There is more and more software coming out which requires at least a 386SX to run. Getting a 286-based computer sticks you with obsolete technology.

Besides the choice of microprocessor, an important decision is selecting the right monitor (display). The current official standard is VGA (Video Graphics Array), but a souped up version called Super VGA is the most popular now. However, there are many variations and quality levels. The parameters to consider include the number of lines of resolution and dots per line. These specifications include figures such as 640x480 (standard VGA). This means that there are 480 scan lines which make up the display with each line having 640 dots. If you look closely at a computer screen, you will see the individual lines. Super VGA configurations include values of 800x600, 1024x768 and even all the way up to 1280x1024. This last high value is still quite expensive.

Just buying a monitor with high resolution is not enough. You will need a video circuit board to go along with it. If your video board does not provide the high resolution output, your super monitor will not do you much good.

No more time for this installment, but I hope that your interest has been aroused. Stay tuned for more later.

=====

R/O MEDIA
(A Summary of Resources about 8080/Z80 Systems)

The Computer Journal (P.O. Box 535, Lincoln, CA 95648-0535, phone: 800/424-8825 or 916/645-1670; bimonthly, $18/yr.): I noted in issue #22/23 that this prestigious magazine changed editors when founder Art Carlson stepped down. Well, it's done it again. For health reasons, Chris McEwen turned his typesetter's apron over to Bill Kibler with #56. I might note that new editor/publisher Kibler has written for **TCJ** for many years. I give **TCJ**'s new address and phone numbers above. Chris' last issue (#55) features Jay Sage's regular "Z-System Corner" with a discussion of "type-3" and "type-4" programs (those that load at addresses **other** than 100hex) and some applications of the latter type. Roger Warren covers remapping disk drives under NZCOM. Paul Chidley presents an update on the YASBEC ("Yet

Another Single-Board Eight-Bit Computer") and a 1 Mbyte memory expansion board for the YASBEC experimenter system. Bill Tishey reviews recent Z-System Software Update Service (ZSUS) releases and updates. Other articles cover fuzzy logic, programming in FORTH, a network protocol, and odds and ends. In Bill's first issue (#56), he prognosticates **TCJ**'s future over the next 10 years. Jay Sage (in his "Z-System Corner") briefly discusses the upcoming ZedFest Europe 1992 and one of the new Z-System emulators for the PC, 4DOS. Tilmann Reh shows how to interface an IDE (AT-bus-style) hard drive to a CPU280 8-bit machine. Walter Rottenkolber presents how to perform modemless file transfers between 1984 and newer Kaypros. Other articles cover more on FORTH programming, embedded systems, and miscellaneous. In issue #57, Jay Sage reviews "language" independent programming (**not** BASIC vs. Pascal or C, but English vs. German or Dutch!). Herbert Johnson introduces the S-100 bus. Rick Swenton presents home automation for CP/M machines using X10 modules. Steven Westlund discusses the most popular file transfer protcols (ASCII; X-, Y-, Z-MODEM; and KERMIT). Terry Hazen shows how to double the clock speed on an AMPRO with the MDISK 1 Mbyte RAMdrive. Lee Bradley briefly reviews **another** Z-System emulator for the PC. Other articles discuss FORTH programming, etc.

COMPUTER MONTHLY (P.O. Box 7062, Atlanta, GA 30357-0062; monthly, $15.95/yr.): This large, shopper-style magazine was the greatest disappointment of the summer. It has **dropped** virtually all coverage of "classic" systems, including CP/M. The only remaining regular column is Nancy Black's "Fearless Computing" with its Commodore flavor, though PC topics are rapidly submerging 8-bit materials. At this point, I don't plan to renew my subscription and will discontinue mention of **CM** with this issue.

Eight Bits and Change (Small Computer Support, 24 E. Cedar St., Newington, CT 0611; 12 issues, $40): Another disappointment this summer was that this newer bimonthly "humour 'Zine" discontinued publication with its Aug/Sept issue (#12). In #10, editor Lee Bradley presents a BASIC program for computing PI to up to 2000 digits. Also here is an update to the Mandelbrot set generator discussed in earlier issues, running the CP/M version of MEX on a PC (under 22NICE), thoughts on the YASBEC, short reviews of the 4DOS Z-System emulator and Z80MU CP/M emulator for the PC, and another installment of Tom Viele's "Hardhacker Chronicles." Issue #11 reviews this year's Trenton Computer Fair. Stephen Griswold describes building the YASBEC. Diane Thome alerts us to the growing menace to fruits, vegetables, and nuts. Bill Kost describes use of a PC virus program designed to **transmit** the vermin; a program package that converts radiation from a computer's CRT into infrared signals to remotely control your TV, VCR, etc.; and an English/Pig Latin or Pig Latin/English translator for ASCII and WordStar files along with a Pig Latin spelling checker. Tom Viele pontificates on the history of computing. Lee Bradley and Bruce Morgen review Z3INSTP. And Al Hathaway and Cam Cotrill briefly discuss BDS Z in a Z-System environment. In the newsletter's last issue, Lee

Bradley reviews SILENT (a utility for suppressing console messages), discusses the "12 ball problem," includes a pair of BBS messages about MYZ80 (one of the Z-System emulators for the PC), and concludes with a menu from Roadkill Cafe. To wrap up this short-lived newsletter, the pricing given above is for the **complete** issue set.

H-SCOOP (2618 Penn Circle, Sheboygan, WI 53081-4250; monthly, $28/yr.): In issue #146, Henry Fale announced "information packages" for older computer systems ('89, '100, '148, '150, etc.) for $10 each. These include specs, basic information, switch and jumper settings, and so forth. In #147, Henry noted (based on information provided by John Brough) that H-25 printer ribbons can be replaced by those for the HP 2631A; sources for the latter include MISCO, Visible, Global, and Best Impressions.

SEBHC Journal (895 Starwick Drive, Ann Arbor, MI 48105; monthly, $24/yr.): The March (vol. VI, no. 8) issue includes an overview on cleanliness and other tips around the computer and the first installment in a series on computing for the "over-55" novice. The latter discusses hardware basics. The April issue has a call for 8-bit articles and the second installment in editor Lenny Geisler's novice series (covering booting and elementary use of HDOS 2.0). In May, the novice series continues with an introduction to FLAGS, SET, HDOS STAND-ALONE, ONECOPY, and SYSGEN; Rick Swenton has a potpourri on PostScript laser printers and X10 home appliance control. The novice series continues in June, dealing with odds and ends before turning to a tutorial on HDOS' line editor, EDIT. The July issue excerpts a column from the San Francisco **Examiner** by John Dvorak discussing collecting vintage (pre-1980) micros; the novice series turns to Benton Harbor BASIC, the interpreter included with HDOS; and Lenny announces acquisition of Pat Swayne's masters for all the software PS: wrote while at HUG. In August, Lenny expands on PS:'s software donation, then continues his exploration of B.H. BASIC in his novice tutorial. And the Sept. issue of **SEBHC Journal** includes thoughts on 8- and 16-bit systems by Rick Swenton and the editor's continuing exploration of B.H. BASIC, presenting a simple text adventure game.

Smaller is Better (CCP/M, c/o Tom Viele, 26 Slater Ave., Norwich, CT 06360; monthly, $15/yr.): The April issue of this newsletter from one of the very few user groups that continues to support CP/M and Z-System, the Connecticut CP/M User's Group, includes an exploration of the YASBEC by Tom Viele. Also by Tom is a discussion of a vitally new concept, Redundant Arrays of Inexpensive Disks (RAID). As you know, working memory in a computer is distributed amongst an array of RAM chips where bytes are stored **bitwise** in each chip. Hence, the bits in any given byte are actually stored in eight different chips. Tom proposes, in order to speed up data transfer, that the same concept be extended to disk drives. So, instead of one to three drives, all computers would have **eight**, whether floppy or hard-disk. (Can you imagine the clatter if you had eight hard-sector drives engaging their head-load solenoids all at once? Can you imagine the storage

you'd need for all those floppies?) The May issue features a review (again by Tom) of Robin Bailey's **The Lost City of Zork** (based on Infocom's Zork adventure software). This issue also reprints a feature article by Gitta Morris from the New York **Times** describing the group. In June, Tom returns with a three-column history of CPUs, extending from the creation of atoms through the creation of assembly language, and a short BASIC program for determining the least common multiple of three numbers. July announces a programming contest to the membership, writing a program to play tic-tac-toe, and an improved version of the least common multiple program from the last issue, submitted by Gerard Weatherby. The August issue contains three articles by editor Tom Viele, a brief "overview" of Z-System, a complaint over the disparaging use of the term "hacker" these days, and presents a "you guess what it does" BASIC program. Also in this issue, Lee Bradley announces the suspension of his **Eight Bits and Change** newsletter (see above). In September, the club newsletter contains a review of MYZ80, a CP/M 2.2/CP/M 3.0/Z-System emulator for PCs, and a discussion of the AVCOMM-BBS, a bulletin board dedicated to aviation. The October issue reprints the first part of a series of articles by Noel Nyman about CP/M on the Commodore 128. In the November issue, part two of Nyman's series is reprinted.

The Z-Letter (Lambda Software Publishing, 720 S. Second St., San Jose, CA 95112-5820; monthly, $15/yr.): The April/May issue is a 20-pager and includes an extended report on the Trenton (NJ) Computer Festival, a discussion of the new format and pricing (forced by editor/publisher Dave Mc-Glone's unemployed condition), Jay Sage's monthly column which presents a ZEX script for loading **backgrounder ii** and updating the date, surveys of magazines and resources that support 8-bit equipment, and concludes with the regular Eagle Computer section. Although the June/July issue has disappeared (given my filing system, not surprising!), the Aug./Sept. one notes the changes at **TCJ** and **EB&C** (described above) and that Dave's software operation, Lambda Software, has acquired the Sound Potentials software library discussed in **Staunch** #25. Sound Potentials goes out of business in December. This issue of **TZL** also includes a number of miscellaneous news tidbits for non-H/Z systems. Jay Sage discusses recursive ARUNZ scripts. Dave describes customizing the Spellbinder word processor for a Kaypro and concludes the issue with his usual survey and Eagle Computer columns.

=====

Patching WordStar 4.0, Part 1 of 2:
Bringing Up the Function and Keypad Keys
By Kirk L. Thompson

Introduction. If you've bought MicroPro International's (now, WordStar International's) last version of the classic word processor for CP/M, WordStar 4.0, from commercial vendors, you undoubtedly noticed some deficiencies in its terminal installation for the H/Z-89. Namely, WordStar chose to ignore implementing the function keys and numeric keypad. Nearly everything else is there that prepares the H/Z-19/89 terminal for use with this

"WP" classic. But you're still forced to resort to a myriad of control-keys that come standard with the package and that you have by now learned to love to hate! This is particularly a problem if you're used to patches you've installed in **earlier** versions of the product. (I might add that the patcher that comes with WS 4.0 is supposed to permit lifting most patches out of 3.0 and 3.3 and installing them in 4.0. But any discussion of that topic is beyond this article's immediate perview.)

But putting up with an unpatched WS 4.0 isn't necessary. All the tools you need to bring up the function keys and the keypad come standard with this WP package. The question becomes one of collecting the information necessary to make the mods. Here, I'm going to discuss how I developed the patches I include with **Staunch's** "custom" version of the word processor. In this first segment, I'll cover bringing up the function keys and keypad; the concluding installment in the next issue will discuss writing to the terminal's 25th line to display function key definitions. Though offered as an "integrated" patch for WS 4.0, the two topics are best described separately. And I wish to thank Trio Co. of Cheektowaga, NY, sole wholesale distributor of WS 4.0, for permission to excerpt sections of the on-disk materials accompanying the word processor.

Resources. There are two primary resources you should have before you attempt patching WS. One of these comes with the package and is the file named PATCH.LST. This is an ASCII file and describes in some depth the structure of WordStar's patch area. When you print it, set your printer for Elite pitch (12 cpi) because many lines exceed 80 characters. This **is** an assembler-oriented listing. But if you don't know assembly language, don't sweat it! You don't need much knowledge of the language to patch the program. Indeed, the one thing this file is invaluable for is as documentation to that patch area. No assembly language expertise is necessary to bring up the function keys and keypad. True, a minimal knowledge is required to install a display on the terminal's 25th line. But if I can do it, so can you!

The second resource is the book recommended by A.E. Thornton in issue #26/27 (p. 5). This is Stuart Bonney's **The New WordStar Customizing Guide** (Wordware Publishing, 1988) and is still available for $24.95 (softcover) through your local bookstore if you can't find it on the used book market. Though oriented mainly toward PC's, it includes a chapter on customizing version 4.0. Some of the tricks I'll describe below are derived from this book. If you're using WordStar 3.0 or 3.3, the book can also be valuable. It includes a discussion of these CP/M-80 editions of the program in an appendix.

If you can't locate this book, you **must** read the 17-page Appendix C in the WS 4.0 manual to introduce you to patching. Then closely follow my discussion below.

There are also some secondary resources you should have at your fingertips. One of these is the operations manual for your computer. Of the things we need information on, most important are the ESCape codes the terminal transmits when the function and keypad keys are pressed. The operations manual supplies this in either section 11 or 12,

depending on whether you have an '89A or the older '89, respectively. You may also need a chart that converts ASCII characters to hexadecimal or a utility program that can do the same thing. If you plan to expand on or to vary my installation, you'll definitely need a conversion chart because most inputs I discuss will be in hex. The patching utility included with WS 4.0 (WSCHANGE.COM) expects hex for most inputs. Bonney's book includes such a table as Appendix A.

WS 4.0's Patch Area. When you send PATCH.LST to your printer, you'll get a 32-page document laid out much like a .LST file from an assembler. That is, hexadecimal numbers occupy the first 16 columns on each page and the remainder of the page is an assembler listing with extensive commenting. Our interest will **mainly** be in this embedded documentation; more on that below.

After you've printed PATCH.LST, there's one other thing you should send to your printer. This is a dump of the **actual** patch area in WS 4.0. But you should do this **after** you've installed the terminal and printer with WINSTALL.COM. Remember, too, that printer installation is a **two-step** process; WINSTALL selects the printer you'll be using, but you also need to run WSCHANGE to install the printer driver(s) you'll need. Refer to p. xxii at the end of the "Welcome" section in the WS 4.0 manual and to the printer driver section (p. 225-7) in the "Reference Guide."

Once you've installed your terminal and printer driver(s), you're ready to dump the patch area to disk and then print it. If you plan to develop your own patches, I strongly recommend creating and editing this dump; it makes life so much easier. So call WSCHANGE from the system prompt and follow the menu system and prompts from main-menu option "E" through sub-menu option "B" (Save settings). When asked for the file name, be sure you give a new one so you don't overwrite something already on disk. Also specify that you want the **entire** patch area. Once you've exited from WSCHANGE, print this file, using Elite pitch, as you did for PATCH.LST, to give you a hardcopy reference.

Planning Key Usage. From the start, you'll have to resign yourself to using at least **some** of WS's one- and two-key-stroke commands. WS 4.0 has over 100 individual control-key sequences. There simply aren't enough programmable keys on the stock '19/89 to do everything! (If you have Lee Hart's "SuperSet" terminal upgrade, the number of available combinations proliferates like rabbits, so you should also exercise greater care when planning which keys do what. But even then, the keys available don't approach the count in WS.) However, it **is** possible to "maximize" those available on the stock machine by setting up the keypad in "alternate shifted" mode. What this means is that the numeric keypad keys from "1" through "9" will send codes to the computer when the SHIFT key is depressed that are different from those when the SHIFT key isn't. The keypad "0", period, and ENTER keys don't respond to the SHIFT key, however. But together with the special function keys across the top of the keyboard and the unshifted keypad, that gives 32 useable keys. (The ERASE key is the only special function

key on the stock keyboard that responds with a different code when the SHIFT key is depressed; it does this no matter what mode the keypad is set for.)

Another issue to consider is whether to use Heath's terminal sequences or the ANSI ones "hidden" in the terminal firmware (ROM) by keying "ESCape <". For many functions, the ANSI codes are generally one or two characters **longer** than are Heath's. But even though that's not the case for the codes transmitted by the keypad (where much of the patch is concentrated), I decided to stick with Heath's. My main reason is because they'll conserve some space, particularly when setting up the function keys across the top of the keyboard. Further, most of the third-party reference materials about the terminal ESCape sequences exclusively document **only** Heath's.

The other side to this planning function is to settle on which keys do what. This, admittedly, is somewhat arbitrary and you'll find my selection given in Table 1 [below]. I've attempted to include the most frequently-used functions, particularly those that normally involve two keystrokes in WS. I've also attempted to keep functions commonly located on certain keys in other word processors on those same keys, but this hasn't always been possible. Further, I've tried to relate unshifted and shifted functions on the same keys of the

keypad. But I haven't done **anything** with five (5) keys (BREAK, SHIFT-ERASE, SHIFT-7, SHIFT-3, and SHIFT-1). That's not because I couldn't have come up with something for them to do! Rather, this presents you with the opportunity to extend my customizing effort if you choose.

If you're designing your own patches, I strongly recommend constructing a table similar to Table 1. The first three columns given there are self-explanatory. So the remainder to the right deserve some exposition. The leftmost "#" is the number of bytes (in hexadecimal) of the code sent by the '89's terminal when the relevant function or keypad key to the extreme left is pressed. The following two or three hex bytes are an "ESCape" character followed by any trailing characters in the sequence. The next "#" is the number of bytes (again in hex) of the WS function we want this sequence to translate to. Then follows the WS function in hex. Finally, the right-most column is the total number of bytes in the translation group, **including** the two byte-counts. And I recommend doing this not only because a table of this kind assists in the design process. Once done, it can be used during data **entry** to the patch file.

Designing the Patch. Once you've settled on the keys you want to install, the next portion of the

Table 1: Function and Keypad Patches

Key	Definition	WS	#	Hex bytes	#	Hex bytes	Total bytes
-->	Cursor right	^D	02	1B,43	01	04	5
shft-->	Word right	^F	03	1B,3F,76	01	06	6
<--	Cursor left	^S	02	1B,44	01	13	5
shft<--	Word left	^A	03	1B,3F,74	01	01	6
up-arrw	Cursor up	^E	02	1B,41	01	05	5
shft-up	Scroll up	^W	03	1B,3F,78	01	17	6
dwn-arr	Cursor down	^X	02	1B,42	01	18	5
shft-dwn	Scroll down	^Z	03	1B,3F,72	01	1A	6
IC	Insert on/off	^V	02	1B,40	01	16	5
IL	Insert line	^N	02	1B,4C	01	0E	5
DL	Delete line	^Y	02	1B,4D	01	19	5
DC	Del. line right	^QY	02	1B,4E	02	11,59	6
shft-DC	Del. line left	^QDel	03	1B,3F,79	02	11,7F	7
HOME	Top of scrn lft	^QE	02	1B,48	02	11,45	6
shftHOME	Bot.of scrn rght	^QX	03	1B,3F,75	02	11,58	7
keypd-0	Prev.cursor pos	^QP	03	1B,3F,70	02	11,50	7
keypd-.	Center line	^OC	03	1B,3F,6E	02	0F,43	7
keypd-ENTER	Bot.of file	^QC	03	1B,3F,4D	02	11,43	7
f1	Find	^QF	02	1B,53	02	11,46	6
f2	Find/replace	^QA	02	1B,54	02	11,41	6
f3	Top of file	^QR	02	1B,55	02	11,52	6
f4	Page up	^R	02	1B,56	01	12	5
f5	Page down	^C	02	1B,57	01	03	5
ERASE	Cancel	^U	02	1B,4A	01	15	5
Blue	Begin blk mrkr	^KB	02	1B,50	02	0B,42	6
Red	End blk mrkr	^KK	02	1B,51	02	0B,4B	6
White	Exit/Save/Menu	^KD	02	1B,52	02	0B,44	6

Total bytes 157

process is to decide where in the patch area they go. Actually, the structure of the function key patch table is described just above a label called "FUNDLY::" on that hardcopy of PATCH.LST you've made. In your dump, this is roughly on p. 3. The beginning of this section, excluding the plentiful comments above and to the left, looks like this:

```
029B              FUNDLY::
029B 00                     DB      0

029C 01                     DB      CTRLA

029D              FUNTBL::
029D 03                     DB      3
029E 01 40 0D               DB      CTRLA,'',CR
02A1 02                     DB      2
02A2 0B 42                  DB      CTRLK,'B'

02A4 03                     DB      3
02A5 01 41 0D               DB      CTRLA,'A',CR
02A8 02                     DB      2
02A9 0B 4B                  DB      CTRLK,'K'
```

and so forth. Actually, you also need to change some things just below the FUNDLY:: label **as well as** install the key translation table.

The comments (not reproduced here) to the left of the two bytes immediately below this label indicate that FUNDLY:: includes two items that **must** be changed if any key translation is to occur. The first of these is a "delay" between the bursts of characters produced when the function keys are pressed. This must be some value **other** than the "00" given for the first byte above. After some thought, I settled on a value of 05 hex for my "stock" '89 and haven't encountered any problems with it. If you've "sped up" your terminal, that is, increased the baud rate between terminal and computer sections, you can probably **decrease** this value.

The second item in FUNDLY:: is the character that introduces each burst. In PATCH.LST, this is defined as "ctrl-A"; of course, for the '19/89 we want "ESCape", so that will have to change to 18 hex.
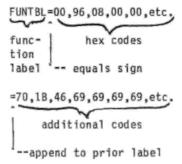
The following label, "FUNTBL::", is the standard location in WS 4.0 for function key definitions. But I noticed one serious wrinkle early on in my patch development. This section is limited to 128 bytes. You've undoubtedly already noticed that the keys I planned to install occupy a minimum of 157 bytes as given in Table 1! If you should want to implement the full complement of 32, there's no way you could possibly fit them into this area.

Another consideration I had was **where** to install the code necessary to write labels for the terminal function keys on the screen's 25th line. My ultimate solution was to **relocate** the function key table and use **this** area for my on-screen key labels. (As it turned out this was a fortuitous decision; the function label occupies most of this area as you'll see.) This is possible because WS includes "undefined" patch areas as well as ones where it expects to find specific information. Two of the former are called "MORPAT::" and "CRTPAT::" and occur around p. 14 on your dump of PATCH.LST. But those two combined only contain 128 bytes and I

disliked the comment in the listing that CRTPAT:: might be used by WINSTALL.COM, the installation program. Further, I wanted to "reserve" MORPAT:: for your use. So I moved the entire translation table to an area called "EXTRA::" at the very top of the patch area. EXTRA:: can hold up to 512 bytes, more than adequate for the task at hand. (If you're patching for use with "SuperSet", here would be the place to install an extended table.) The only thing you lose is half a K of buffer space for text you edit.

Patch Installation. As I implied above, I set up the patch by editing a dump of the patch area. Once I was satisfied with that, I installed the patch with WSCHANGE. Figure 1 [on the following page] presents the **entire** patch, including the mods to write the function key labels on the terminal screen. The **entire** patch area you dumped to disk is considerably larger. Once you have the patch file on disk, you have some editing to do. So load the patch file into WS as a **non-document** and put up with all those control-codes just this once. A folded "Command Card" for your reference is included with the package.

The format of the patch file is straightforward. For example, FUNTBL:: is composed of:

```
FUNTBL=00,96,08,00,00,etc.
   ^              ^
func-     |      hex codes
tion      |
label     '-- equals sign

=70,1B,46,69,69,69,69,etc.
 ^              ^
       additional codes
   |
   |
   '--append to prior label
```

From the figure, you can see that those areas which are large, such as EXTRA::, have a lot of appending equal signs. One thing you **shouldn't** do, as you do your own editing, is to chop 00 hexes from the end of labeled areas. Those will serve as your guide to the size to each area.

Now edit out all the labels **not** included in Figure 1 and shorten the longer lines to something more convenient. The latter is easily accomplished by inserting a **hard** carriage return at a comma and replacing that comma with an equals sign at the left margin. That's what I've done in the figure to get less than 80 characters to a line. The patching program (WSCHANGE) doesn't particularly care how long a line is.

Once that's done, go back and key in the codes for the patch. I've already discussed the translation table in EXTRA::, the two mods to FUNDLY::, and some of what's going on in FUNTBL::. The first **three** (3) bytes in the last are essential for the translation table to function correctly. These bytes are an unconditional "jump" to EXTRA::. The balance of this area is the function key label patch and I'll discuss that, as I've noted, in the next issue. But there are some other things that must be covered so you understand what else is going on here.

These additional changes mainly constitute setting up the **terminal**. And I'll discuss them in the

order they appear in Figure 1. However, different areas have different functions; some are intended to contain characters to be transmitted to the terminal, others are for jumping to or calling machine-language routines, and some can be use for **either** purpose. So their structures will be different.

The first of these areas in Table 1 is ERASCR::, one that apparently is called anytime the screen is totally blanked and the cursor homed (moved to the upper left-hand corner). This label is typical of an area whose primary purpose is to store characters for transmission and originally it looked like this

---

Figure 1: WordStar 4.0 Patch

```
FUNDLY=05,1B
FUNTBL=00,96,08,00,00,21,AB,02,CD,83,02,C9,57,1B,6A,1B,78,31,1B,59,38
=20,1B,70,1B,46,69,69,69,69,69,46
=49,4E,44,69,69,69,52,45,50,4C,69,69,69,54,4F,50,69,69,69,69,50,47,20
=55,50,69,69,50,47,20,44,4E
=69,69,43,41,4E,43,69,69,69,42,45,47,20,42,4B,69,45,4E,44,20,42,4B,69
=45,58,49,54,69,69,69,1B,47
=1B,71,1B,6B,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00
ERASCR=0C,1B,48,1B,4A,1B,77,1B,74,00,1B,3D,00,00,00,00
TRMINI=FF,86,03,00,00
TRMUNI=02,1B,7A,00,00
INISUB=C3,A2,02
UNISUB=00,00,C9
INITID=28,50,61,74,63,68,65,64,20,66,6F,72,20,66,75,6E,63,74,69,6F,6E,2F
=6B,65,79,70,61,64,2F,32,35,74,68
=20,6C,69,6E,65,29,00
RAM1ST=96,0A
EXTRA=02,1B,43,01,04,02,1B,44,01,13,03,1B,3F,76,01,06,03,1B,3F,74,01,01
=02,1B,41,01,05,03,1B,3F,78,01,17
=02,1B,42,01,18,03,1B,3F,72,01,1A,02,1B,40,01,16,02,1B,4C,01,0E
=02,1B,4D,01,19,02,1B,4E,02,11,59
=03,1B,3F,79,02,11,7F,02,1B,48,02,11,45,03,1B,3F,75,02,11,58
=03,1B,3F,70,02,11,50,03,1B,3F,6E,02,0F,43
=03,1B,3F,4D,02,11,43,02,1B,53,02,11,46,02,1B,54,02,11,41
=02,1B,55,02,11,52,02,1B,56,01,12
=02,1B,57,01,03,02,1B,4A,01,15,02,1B,50,02,0B,42,02,1B,51,02,0B,4B
=02,1B,52,02,0B,44,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
=00,00,00,00,00,00,00,00,00
```

after installation with WINSTALL:

ERASCR=06,1B,48,1B,4A,1B,77,00,00,etc.

The first hex digit in the area is the **number of characters** that is to be sent to the terminal; there are six (6) in this instance. The installed values are ESCape H, ESCape J, and ESCape w. These are the terminal sequences for homing the cursor, erasing to the end of the screen, and discarding any characters beyond the 80th on a line, respectively. Because I was cramped for space elsewhere and also because this segment isn't used for much more than clearing the screen when WordStar jumps between the sign-on menu and the edit screen, I decided to use it to set up the keypad. And as you'll see in the next paragraph, one other function label "jumps" to this one. So I **added** two terminal sequences, ESCape t and ESCape =, to activate keypad shifted and alternate modes and followed **each** with a "null" byte (00 hex) to serve as a delay. Since the entire sequence now numbers 12 characters, I increased the introductory hex digit to OC.

I had better add, too, that while preparing this article, I discovered a "bug" in this area in the patch file I provide with WS 4.0. As supplied, the first hex digit for ERASCR:: is "1C". It **should** be the "OC" noted above and given in Figure 1. If you've purchased WS 4.0 from me, I've either notified you of the patch bug by postcard, or in recent shipments fixed it in the distribution files.

TRMINI::, the next area, is intended for storing a terminal initialization string. This string is executed **only** when WordStar signs on. But the area itself isn't very large (four bytes, not including an introductory count digit) and the default "value," as it were, is a "jump" to ERASCR::! This is designated by the initial FF hex as illustrated in Figure 1 and must be followed by a two-character memory address in Intel's "standard" low-signifi-cant-byte/high-significant-byte sequence. The struc-ture of this "jump" is also typical of what WordStar 4.0 expects to see when "told" to jump and the area is dual-functional. Actually this served my purposes as was. So, although I haven't done anything with this, I've left it in the patch just in case you need it.

TRMUNI:: is a sequence that's sent to the terminal when WordStar exits back to the operating system. Since, like TRMINI::, the area is only four bytes long, I settled on the simplest reset sequence possible, ESCape z.

INISUB:: is an area intended for an initializa-tion routine **different** than TRMINI::. And given the three-byte size of the thing, about the best that will fit into it is a jump instruction and an address. And that's what you see there in Figure 1. This "jump" is different than the one in TRMINI, however. This is the point where I write the terminal's 25th line with the function key labels. And since I'm jumping into the **middle** of FUNTBL::, I opted to use the 8080 MPU's unconditional jump instruction, C3 hex, which has the mnemonic name, JMP. The following two bytes in INISUB:: are the address for the "21" you see immediately following the two null bytes near the beginning of FUNTBL::.

The next area, UNISUB::, is another small area for an un-initialization subroutine address. This one I left untouched. And I should note that the "C9" in this area in Table 1 should be left as is unless you actually use this area. C9 hex is the 8080's "RETurn" instruction.

The largest of these miscellaneous patch areas we're examining is INITID::. This is an area where you can set up a character string that will display on the terminal while WS 4.0 signs on. The main limitation is the **length** of the string, which is 39 characters. The fortieth character in this area **must** be a null (0 hex). The string I've embedded reads: "(Patched for function/keypad/25th line)". If you prefer, you can use ASCII characters and place them between single quotes. As a side note, there's also an area like this one for identifying your printer at label PRNID::.

Finally, the last label in the patch is RAM1ST::. This comes with a "standard" value **different** from that given in Table 1. Originally, this read

RAM1ST=96,08

This address is the **lower** address of your text buffer in WS 4.0. Since, however, I decided to use EXTRA::, I've increased this address by 512 (200 hex) bytes to provide room (and more) for the translation table. My patch leaves this area at its maximum of 512 bytes. If you wish to recover some of this for your files and don't anticipate doing anything else with this table, you can recover **half** the area by changing this value to "96,09".

Looking Toward Labeling the Function Keys. Next issue, I'll describe how I designed the patch to write function-key labels on the terminal's 25th line. And if you **already** have WS 4.0 and would like this patch on disk, you can have it at no charge. Just send me a formatted disk and a postage-prepaid return mailer; I'll tranfer it for you.

Finally, I must note that **all** the copies of WS 4.0 I had are now gone! However, if you want one at something cheaper than the $150 retail price, I'll take "reservations." **Don't** send money! Just indicate whether you're interested. When I have enough reservations for a case (seven copies), I'll order one and ask you to send $90 (including U.P.S. shipping in the contiguous 48) for yours.

=====

THE LINKAGE LOADER
(A column of reader-furnished routines)

In issue #24 (p. 1), Bob Phillips of Charlotte, MI, presented an HDOS MBASIC program for printing text files on both sides of the paper. One limita-tion in it was that the printed file couldn't include the formfeed character (12 decimal, ctrl-L). However, a few simple changes to Bob's program permits the embedding of formfeeds in the text to control paging, rather than relying on the program itself to count lines.

The changes I made were as follows. Alterations **within** a line are boldfaced; new lines of code have only the line number boldfaced:

```
70 CLEAR 5300:DEFINT A-Z
140 LP$="LP:"
```

```
170 LASTLNE = 66
405 IF INSTR(1,LI$(ODDPAGE),FOMFEED$) > 0 THEN@
    FFLNE = ODDPAGE
430 FOR PRNTPAGE = FIRSTLNE TO FFLNE
460 'PRINT #2, FOMFEED$
535 IF INSTR(1,TRASH$,FOMFEED$) > 0 THEN@
    TRASHPAGE = LASTLNE
815 IF INSTR(1,TRASH$,FOMFEED$) > 0 THEN@
    TRASHPAGE = LASTLNE
885 IF INSTR(1,LI$(EVENPAGE),FOMFEED$) > 0 THEN@
    FFLNE = EVENPAGE
910 FOR PRNTPAGE = FIRSTLNE TO FFLNE
940 'PRINT #2, FOMFEED$
```

By way of comment, the CLEAR statement in line 70 reserves enough room in memory for any text the program is likely to see; before I made this change, the program bombed! Line 140 sets up the correct HDOS printer device driver for my system. The change to line 170 sets the read and write FOR..NEXT loops occurring later to read/write a **full** page of text on a page formatted for six lines per vertical inch; if your preference runs to 8 lines per inch, set this to 88 instead. If you do the latter, you should also reserve more memory in line 70, on the order of 7100 bytes.

Bob's program starts its actual work in line 390 (see issue #24 for the original program listing). What I had to do in each of the relevant FOR..NEXT loops was to **detect** the presence of a formfeed in a line of the input file, then use that to control output to the printer. The input phase is given in lines 405, 535, 815, and 885. Here I'm using MBASIC's INSTR() statement to detect a formfeed **anywhere** in the input lines for **both** odd- and even-numbered pages. If the character is there (lines 405 and 885), I **save** that line number in the variable, FFLNE, when a page (odd or even) is to be printed. When a page (even or odd) is to be thrown away (lines 535 and 815), the conditional with the INSTR() statement sets up an exit from the loop the next time the NEXT statement is encountered. I specifically wanted to avoid GOTOs in this latter situation because it would clutter up the program stack (where GOSUB..RETURN and FOR..NEXT line references are stored). If GOTOs are used too frequently to exit FOR..NEXT loops or GOSUB..RE-TURNs, the stack will fill up and the program get lost. I've had that happen!

During the printout loops (lines 430 and 910), that new variable, FFLNE, controls what's sent to the printer. And I commented out lines 460 and 940. All my files that contain embedded formfeeds **also** include one at the end of the file, so there's no reason to throw in extraneous ones except at the very end of the printing process. In effect, the **file** controls all of the page feeds.

Of course, a program like this isn't intended for short files. But it shines on long ones where you wish to conserve paper. Specifically, I tested it on a 134K (535 HDOS sector), 45-pager and it ran like a champ.

=====

Multitasking for Real-Time Response...
Under HDOS?   Part 3 of 4
By David A. Shaw

HANDLING SPEED DIFFERENCES. Each task in a multitasking system will run at its own speed. Earlier in this series [issue #26/27, p. 10], we mentioned a case where characters were being received from a modem at a rate of 120 cps, while the printer in use could only print them at 45 cps. If something isn't done to resolve these speed differences, there will be data loss.

The ultimate source of a stream of data is called the producer. The task that it sends that data to is called the consumer. A good example of a producer is an interrupt service routine which takes in characters at whatever rate they arrive at the I/O port. The consumer in this case would be the task that reads and processes those characters.

If the producer can outpace the consumer, even for a short while, then some kind of buffering scheme is needed between them to take up the slack. If the unit of data moving from producer to consumer is a character, then a character-sized circular buffer of some size can be used. If the unit of data is larger, then it may be appropriate to use a message buffer and a queue.

If you use HDOS, then you routinely use a circular character buffer; HDOS uses one for characters received from the terminal. A circular buffer is a contiguous area of memory some number of bytes long, addressed by two indexes or pointers; one is used to put characters in and one is used to take characters out. The buffer is empty when the in and out pointers are equal. It has data in it if they are not equal, and it is full if the insertion of one more character would cause the pointers to become equal. The buffer is called circular because the pointers wrap around from the end of the buffer to the beginning, in a kind of a circular motion.

When you use a circular buffer between two tasks that use SWAP [see issue #26/27, p 9], you don't have to use any particular technique to put data in or take it out. However, when you put one between an interrupt service routine and a task, you have to be careful when accessing the buffer. An interrupt can occur at any time, even when the task is retrieving data from the buffer. The task won't know that the interrupt occurred. It is important to make sure that the interrupt service routine and the task don't update the same pointers at the same time.

Most programmers will disable interrupts when accessing a circular buffer. I avoid that approach when possible because it can have quite an impact on system throughput. You use interrupts to give immediate service to incoming characters, and if you disable them frequently you are working against this goal. Instead, make sure that the task writing to the input side of the buffer modifies only the input pointer, and then only after the character is inserted into the buffer. Likewise, make sure the task reading the buffer modifies only the output pointer, and only after the character has been read.

Correct sizing of the buffer is a trick. If you expect a constant stream of characters to arrive at an interface, the tasks reading and processing those characters had better be fast enough to keep up, or you need a good way to slow down, or flow control, the incoming data stream. The buffer size used should reflect the latency of the system, that is, the amount of time it takes to run all tasks. I usually use a small size for keyboard or modem input, say 16 bytes, as I fully expect to keep up

with the incoming data. For the printer, however, I generally reserve all memory that remains after everything else is allocated. In one case, I had a 32 K/byte printer buffer!

Flow control is important, however, even if you have a huge buffer. With my printer buffer, I would let it fill to within 256 bytes of full, then I would cause the sending task to send an XOFF character (Control-S) to the modem interface to cause the connected host to stop sending for a while. (It's important to stop the host before the buffer is completely empty in case the host cannot respond immediately. I stopped within 256 bytes in this case because it was convenient.) When the buffer had more than 512 bytes of free space, I would cause the sending task to send XON (Control-Q) to turn the host back on. You should have a healthy gap between the "off" and the "on" values to prevent your program from oscillating between stopping and starting data flow.

If you are sending blocks of data or other data structures from task to task, you need to pass buffers through a queue. You start with a buffer pool allocated in your initialization routine. Tasks which need a buffer fetch one from the pool. After placing data into the buffer, they send it to the next task by chaining it to the end of a queue that is read by that routine.

The code for initialization of a buffer pool, which is just a queue, or linked list, full of empty buffers, and two routines, Q.GET and Q.PUT (which are used to take buffers off a queue and put them on another queue) is illustrated in EXAMPLE.ASM. These routines depend on a few data structures:

```
* The Queue Control Block Structure
*
          ORG  0
Q.COUNT   DS   1         Count (max 255)
Q.FIRST   DS   2         Pointer to front
Q.LAST    DS   2         Pointer to back
QCB       EQU  *         Length of structure

* Queue Entry (Buffer) Structure
*
          ORG  0
QE.NEXT   DS   2         Pointer to next in chain
QE.DATA   DS   ???       (Application specific)
QE        EQU  *         Length of a QE

POOLCNT   EQU  ???       (Application specific)
```

You'll also have to provide a base address for the buffer pool; Q.INIT expects you to define a label named POOLBEG.

With these definitions in place, we can define various queues and initialize a buffer pool. To define a queue:

```
POOL      DS   QCB       Define a buffer pool
PTR.QUE   DS   QCB       Define a printer queue
TERMQUE   DS   QCB       Define a terminal output queue
```

Your initialization routine should zero the Q.COUNT byte in each QCB. After the buffer pool is initialized by calling Q.INIT, we can use the buffers as follows:

```
LOOP      EQU  *
          LXI  H,POOL
          CALL Q.GET     Get a buffer
          CC   SWAP      Empty! Let someone free one...
          JC   LOOP      ...and try again
*
* (HL) contains buffer address.
*
          PUSH H         Save the buffer address

          ...  < fill the buffer >

          POP  D         (DE) = Buffer address
          LXI  H,PTR.QUE
          CALL Q.PUT     Send the buffer to the printer
          JMP  LOOP      ...and loop back
```

The printer task, in the above example, would wait for data to arrive, then send it to the printer:

```
PRINTER   EQU  *
          LXI  H,PTR.QUE
PR.1      EQU  *
          CALL Q.GET     Look for data
          CC   SWAP      Wait if none
          JC   PR.1      ...then try again
          PUSH H         Save buffer address
*
* Find front of data and send it to the printer.
          ...
* Return the buffer to the pool
*
          POP  D         (DE) = Buffer Address
          LXI  H,POOL
          CALL Q.PUT     Chain to back of pool
          JMP  PRINTER   ...and look for more
```

EXAMPLE demonstrates the use of Q.GET and Q.PUT, and contains another useful queue management routine, Q.FLUSH, which is used to completely clean out a queue, returning all buffers to the pool. It also contains Q.REQUE which puts buffers onto the front of the queue rather than the back in those odd instances where this makes sense. Neither of these are actually used by EXAMPLE. [Dave's EXAMPLE files are still available at no charge; send **Staunch** a formatted disk and self-addressed postage-prepaid mailer and I'll tranfer them for you. The files only require 68K. -Ed.]

=====

VENDOR.UPDATE

Changes at H/Z. If you read any of the other Heath/Zenith-related publications, you already know of the changes that have occurred at Heath Co. and Zenith Users Group. So I'll only briefly reprise those here.

HEATH CO. Long-time electronic kit maker, Heath dropped out of the kit and "home automation" business this last summer. The principal reason, according to Heath president Bill Johnson, was the economic change brought about by inexpensive imports from the Far East. Mr. Johnson and I had some correspondence on this subject. If you would like to see it in machine-readable form, let me know. In my queue, once I've caught up on other things, is the preparation of this material (with, of course, Mr.

Johnson's permission) for general circulation.

In any case, Heath is shifting to educational products. Some of these will involve kits, but the latter will only **support** the educational materials. Personally, I would have remained extremely skeptical if Heath had decided to continue distributing its "home automation" line of products. I've had discussions with one of its suppliers and according to it, although Heath's sales were good, the profit margin from that product line wasn't that great. By moving into education, Heath at least plays to one of its strengths. The materials (documentation and home-study courses) that it produced independently of ZDS were almost always impressive. Though the acquisition by Zenith certainly eroded the quality of the materials produced over the last decade.

Still, this marks the end of an era. Sophisticated electronic kits, with few exceptions (the most recent issue [#11] of Paul Herman's **Z-100 LifeLine** mentions the possibility that ECI--more on it below--may offer a 486-based PC clone), are a thing of the past. Further, like many others, I believe there's still an electronic kit market out there. But now there's no one to service it.

ZUG. Although forewarned by Henry Fale's **H-SCOOP** #147 just a few days before I received it, a letter dated 8 June, 1992 arrived from ZDS announcing the termination of **REMark** magazine and, implicitly, the demise of ZUG. Only one other thing out of this situation pleased me: the arrival of a check from ZDS several weeks later to refund the remainder of my subscription. At least **REMark** didn't fizzle out like Sextant Publishing did in mid '89.

ECI. But you can't keep a good group down, so to speak. The core management of ZUG, namely Bob Ellerton, Jim Buszkiewicz, and Pat Swayne, have resurfaced with their own prospective organization, Electronic Concepts Inc. (ECI). With it, they plan to continue "the tradition of providing independent support to a well-established user community." In the works is an on-disk newsletter (described as such in a recent letter to me from editor Pat Swayne), a BBS (offering downloadable software, messaging, conferencing, etc.), and miscellaneous services (such as the "kit" system mentioned above) and group purchases. From our perspective, the principal "downer" is that ECI's orientation will be toward PCs.

One benefit, however, has been the "liberation" of ZUG's old 8-bit software library. Pat Swayne (presumably) obtained Heath's permission to place the entire collection in the public domain and he, in turn, passed his originals to **SEBHC Journal** editor/publisher, Leonard Geisler. As I mention elsewhere in this issue, Lenny recently announced their acquisition and intent to share them with **STAUNCH.** As I write this, Lenny has organized and sent me the first block of ZUG's material, all HDOS software and covering **60** single-sided disks. The CP/M material is to follow. I'll begin listing this stuff in the next issue. I might add, further, that there's **more** here than what ZUG included in its regular price lists in **REMark**! And I must thank **both** Lenny and PS: for their generosity.

In any case, initial subscriptions to ECI's newsletter are $25. For more information, contact

the "new" HUG at:

Electronic Concepts Inc. / P.O. Box 348 / St. Joseph, MI 49085 / voice: (616) 849-0491, data: (219) 654-3210

The Printer Works. If you have a Xerox/Diablo daisywheel printer and/or are interested in PostScript-compatible laser printers, I suggest you contact The Printer Works. This outfit supplies parts and accessories for the old daisywheel printers and for Canon SX-based, Adobe PostScript- and Microsoft TrueImage-compatible laser printers, parts, and accessories. Given the problems I had with the factory trying to get my old Diablo 630 fixed, this company should be the source of choice for parts for Xerox/Diablo's printers. It also provides upgrades for older Apple and HP laser printers. For more information, contact:

The Printer Works / 3481 Arden Rd. / Hayward, CA 94545 / (800) 235-6116

Z-AT. The number of emulators for running CP/M and/or Z-System software on PC clones is proliferating. I recently received information on a product called MYZ80, developed by an Australian, and available for downloading or via mail from a state-side source. The package is targetted for AT-class PC clones and emulates Zilog's Z80. Because this is shareware, registration with the author is US$23.50. And while you can run CP/M programs "out of the box," as it were, the U.S. distributor recommends Sage Microsystems' NZCOM to bring up Z-System. If you expect to move to PCs and were worried about porting all of your CP/M programs and files to the new system, there appear to be an increasing number of ways to avoid trashing everything you now have. For details on MYZ80, contact:

Lee Bradley / Small Computer Support / 24 East Cedar St. / Newington, CT 06111

or download the package (named MYZ80100.ZIP) from Z-Node #12, 203/665-1100. The file only occupies 58K.

Micronics Technology (410 Bellehurst Dr. / Montgomery, AL 36109 / voice & FAX: (205) 244-1597, BBS: (205) 244-0192) "Thanks for your recent letter. I still support the WIN89, but my available time will dwindle to about 1 day a month after January 1993. I'll be in school for five months starting in January and moving to a new assignment in June. After I find out what my new job is, I'll look into revitalizing Micronics.

"I am in the process of closing out my stock and reducing prices on all Micronics products. The WIN89 kit is now $249. The kit includes everything you need to install a 20 meg hard disk in the H-89 for CP/M. I have two complete kits left. Rich Musgrave is purchasing a WIN89 interface card, so HDOS support may be available soon. The interface card and software is $99. It requires a WD1003-compatible PC controller and a 10 or 20 meg hard disk. The interface and controller is $149.

"I also have the following available (prices include shipping):

    Magnolia floppy controller (5" and 8") $50
    C.D.R. floppy controller (5" and 8") $50
    Z-100 8" drive and case (1/2 height) $60

...My BBS will have the latest information in the
bargain section. Please note the new address above.
After 12/31/92 my voice phone will be 205/244-0112.
Also all checks need to be made out to Darrel Pelan.
I no longer have a business checking account or
ability to take VISA cards.

    "Thanks for your support to the 8-bit community.
I wish you and all the active 8 bit users well."

=====

Using T/MAKER, Pt. 1:
Introduction
By Kirk L. Thompson

Background. Heretofore, when anyone mentioned
"integrated software" to me, I hadn't thought of
CP/M. Rather, the term brought to mind APPLEWORKS,
the combination word processor/database/spreadsheet
program for the Apple II series, and SYMPHONY, a
similar product Lotus Development built around a
stripped-down version of 1-2-3 for the PC. Or even,
as Lee Hart recently reminded me, CONTEXT MBA,
though he noted it was originally developed for CP/M
whereas I encountered it some 10 years ago on a Hew-
lett-Packard PC semi-clone.

    So it came as some surprise to discover Lee's
mini-review of T/MAKER for CP/M in **H-SCOOP** #136 and
his follow-up notes on configuring the package in
#139. After some hesitation, I bought it to see what
it could do. What Lee (later on) and I will be doing
in this series is to discuss the capabilities of
this package in something of an "integrated" manner.
First off, though, I should describe just what this
package does.

Integration. So what is T/MAKER? It's an "inte-
grated" package that can perform the following tasks
for you:

    o   Limited file management
    o   Word processing
    o   Spell checking
    o   Spreadsheet
    o   Database flat-file management
    o   List processing, including sorting
    o   Data transfer between files
    o   Bar charting
    o   Programming to automate the above tasks

Included with the package are online help and exten-
sive printed documentation.

    The software, as it comes on the distribution
disk set, occupies some **700K**! This includes files
for a small, hands-on tutorial; programming exam-
ples; help files; and the core program with its
overlays.

    And this package uses a **lot** of overlays (almost
30 in the basic package), so requires considerable
disk storage. For example, I have the core system
spread over three 48-tpi, soft-sector floppies.
Accessing overlays on different disks requires disk
swapping in drive A:. So, because most commands are
separate overlays, disk swapping could be something
you do a lot unless you have a hard-drive. Indeed,

T/MAKER would be ideal on a winchester. And though
runnable on standard hard-sector, disk swapping
could become a major headache. So, to make effective
use of this package, I strongly recommend two drives
and extended disk storage **per** drive. If running
hard-sector, you should have **at least** double-sided
drives, if not 96-tpi.

    However, unlike separate applications, the
strength of this package (and similar software for
other systems) is that you can easily **blend** differ-
ent types of functions into one document. For
example, say you need a spreadsheet **inside** a
printed report. With separate word processor and
spreadsheet programs (such as WordStar and
SuperCalc), you would have to prepare the
spreadsheet with SuperCalc, save it as an ASCII file
on disk, then merge that file into your document
with WordStar. Further, if you needed to
periodically update the report and embedded
spreadsheet, you'd have to repeat the process each
time. That could become a major hassle if you need
to do it frequently.

    With T/MAKER, you no longer have to do that. You
can embed the spreadsheet **directly** inside your re-
port. When you need to update the document, all you
need do in the spreadsheet is to **replace** the numer-
ical inputs, recalculate it, and you're done!

Documentation. The printed manuals that come with
the package are impressive. The main manual includes
some 475 pages in softcover format. It contains a
copy of the tutorial at the beginning and a refer-
ence guide to the package at the end. These latter
two items are also separate booklets of 56 and 47
pages respectively to make access easier and are
typeset. The remainder of the main manual appears to
have been prepared with the word processor-portion
of T/MAKER on a dot matrix printer. It is, however,
very readable (mainly because of some photoreduc-
tion), **quite thorough**, and detailed. Each of the
package's capabilities is discussed in a separate
section. There's also a serviceable index.

    The on-line help facility is hardly mentioned in
the documentation, but is one of those overlay
commands mentioned above. In that sense, it must be
called from the "WHAT NEXT?" (command-entry) prompt
so isn't context-sensitive. However, it provides
some information not available elsewhere--such as
most of the function and keypad key assignments set
up by Lee for the '89--and summaries of almost all
the available commands. Since it is menu- and
command-driven, once you get into it, the help
system is easy to access and use.

Miscellaneous. I noted above that T/MAKER has
"limited" file management capabilities. In other
words, it is **not** a replacement for PIP. It's com-
petence is in handling those data files you create
while using the package. That is, you can create,
retrieve, delete, rename, and list individual files;
display a disk directory; insert or overlay one file
with another; and truncate files at either top or
bottom. Another pair of commands can be used to
prepare text files before they are to be processed
by programs other than T/MAKER. This is necessary
because T/MAKER is rather idiosyncratic when it
comes to TABs. Data disk drives may also be reset
from the command prompt, though this is not neces-

sary when swapping program disks in drive A:.

The built-in spelling checker recognizes some 52,000 words, split between two dictionaries. The checker can be used to proof either entire documents or to check individual words; the latter is accomplished at the command prompt. You may, of course, add new words to the dictionaries.

Coming Topics. As I remarked above, the intent of this series is to present an "integrated" tutorial to the package. Suppose, for the sake of illustration, that you've just been elected secretary/-treasurer of your local computer users' group. And to aid in computerizing the records you'll need, you've also just bought T/MAKER. The next segment will discuss setting up T/MAKER for the first time, use of the function keys on the '89, and customizing the package to work with your specific printer. So see you then.

Source: Quikdata, Inc.
             2618 Penn Circle
             Sheboygan, WI 53081-4250
             Price - $59 plus $5 s/h
             Soft-sector only!

If you need **hard**-sector, **Staunch** can do the media conversion for you. Just bear my cautionary note above in mind.

=====

A List of Tools Needed to Repair the H-89
By Dan Jerome

GROUP 1: FOR QUALIFIED ELECTRONICS TECHNICIANS
The H-89 schematics and the Operations Manual.
A second H-89 computer (dead or alive) used for parts.
Oscilloscope to observe various waveforms as called out.
Frequency counter.
Digital volt, ohm, current meter.
Soldering iron with about 25-watts capacity.
Large and small heat sinks for protecting solid-state devices.
One each DuPont 1-1/2-inch tapered polyester **soft** paintbrush or the equivalent.
Large magnifying glass for checking device part numbers.
Loctite for securing loose chassis parts and top cover.
Small can of TV Tuner Cleaner with light lubricant.
Digital logic probe.
Digital logic pulser.
Clip-on integrated circuit test connectors.
Test fixture to support the boards under investigation.

GROUP 2: GENERAL PRACTICE
Digital volt, ohm, current meter. Analog meters are okay but are more difficult to read measurements accurately!
Soldering iron. NOTE: Do not use a large soldering gun or the equivalent, as solid-state devices such as transistors, voltage regulators, and

microcircuits are highly heat sensitive and can be easily destroyed.
Large and small heat sinks for protecting solid-state devices.
1/4-inch nut driver.
Set of small screwdrivers.
Large flat-bladed screwdriver to open the case.
Medium sized flat-bladed screwdriver.
Medium sized phillips screwdriver.
Small 5-inch needle nose pliers.
Wire strippers for emergencies.
Standard and long tweezers with positive grip.
One each DuPont 1-1/2-inch tapered polyester **soft** paintbrush or the equivalent. The tapered brush allows you access to tight areas much better than the flat brush. Also, you do not need a cheap brush that leaves bristles behind, as this could be hazardous to your computer.
Loctite to secure chassis parts and top cover, etc.
Large magnifying glass (3-inches or better) to read part numbers, etc.
TV Tuner cleaner and lubricant. Note the light lubricant is preferred because it leaves a lighter residue.

NOTES
1. Save all of the attaching parts, since they are difficult to find on the market. Substitute attaching parts with parts that are equivalent.
2. Procure a second H-89, dead or alive, so that you have a ready source of parts.
3. The majority of parts used on the H-89 are easily obtained from mail order houses such as JDR Microdevices (2233 Samaritan Drive, San Jose, CA 95124) or Jameco. In a pinch, you can try Radio Shack or other local parts jobbers.

[With this segment, the hardware troubleshooting series **temporarily** pauses. It will resume as manuscripts from the technical consultants arrive. -Ed.]